(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 680 654 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
02.09.1998 Bulletin 1998/36

(21) Application number: **94907838.0**

(22) Date of filing: **18.01.1994**

(51) Int Cl.[6]: **G10L 5/04**

(86) International application number:
PCT/US94/00649

(87) International publication number:
WO 94/17518 (04.08.1994 Gazette 1994/18)

(54) **TEXT-TO-SPEECH SYSTEM USING VECTOR QUANTIZATION BASED SPEECH ENCODING/DECODING**

Text-zu-sprache-Uebersetzungssystem unter Verwendung von Sprachcodierung und Decodierung auf der Basis von Vectorquantisierung

SYSTEME DE SYNTHESE VOCALE A CODAGE/DECODAGE DE SIGNAUX VOCAUX BASE SUR LA QUANTIFICATION VECTORIELLE

(84) Designated Contracting States:
DE ES FR GB

(30) Priority: **21.01.1993 US 7191**

(43) Date of publication of application:
08.11.1995 Bulletin 1995/45

(73) Proprietor: **APPLE COMPUTER, INC.**
Cupertino, California 95014 (US)

(72) Inventor: **NARAYAN, Shankar**
Palo Alto, CA 94306 (US)

(74) Representative: **Hughes, Andrea Michelle**
Frank B. Dehn & Co.,
European Patent Attorneys,
179 Queen Victoria Street
London EC4V 4EL (GB)

(56) References cited:
EP-A- 0 515 709
US-A- 4 384 169
WO-A-85/04747
US-A- 4 833 718

EP 0 680 654 B1

D scription

The present invention relates to translating sound segment codes representing speech, or text, in a computer system to synthesized speech; and more particularly to techniques used in such systems for storage and retrieval of speech data.

In text-to-speech systems, stored text in a computer is translated to synthesized speech. As can be appreciated, this kind of system would have wide spread application if it were of reasonable cost. For instance, a text-to-speech system could be used for reviewing electronic mail remotely across a telephone line, by causing the computer storing the electronic mail to synthesize speech representing the electronic mail. Also, such systems could be used for reading to people who are visually impaired. In the word processing context, text-to-speech systems might be used to assist in proofreading a large document.

However in prior art systems which have reasonable cost, the quality of the speech has been relatively poor making it uncomfortable to use or difficult to understand. In order to achieve good quality speech, prior art speech synthesis systems need specialized hardware which is very expensive, and/or a large amount of memory space in the computer system generating the sound.

In text-to-speech systems, an algorithm reviews an input text string, and translates the words in the text string into a sequence of diphones which must be translated into synthesized speech. Also, text-to-speech systems analyze the text based on word type and context to generate intonation control used for adjusting the duration of the sounds and the pitch of the sounds involved in the speech.

Diphones consist of a unit of speech composed of the transition between one sound, or phoneme, and an adjacent sound, or phoneme. Diphones typically start at the center of one phoneme and end at the center of a neighboring phoneme. This preserves the transition between the sounds relatively well.

American English based text-to-speech systems, depending on the particular implementation, use about fifty different sounds referred to as phones. Of these fifty different sounds, the standard language uses about 1800 diphones out of possible 2500 phone pairs. Thus, a text-to-speech system must be capable of reproducing 1800 diphones. To store the speech data directly for each diphone would involve a huge amount of memory. Thus, compression techniques have evolved to limit the amount of memory required for storing the diphones. However, to be successful, the computational complexity of the decoder for decompressing the diphone data must be very low so that the system is capable of running across a broad range of hardware platforms with very high quality reproduction.

Prior art systems which have addressed this problem are described in part in United States Patent No. 8,452,168, entitled COMPRESSION OF STORED WAVE FORMS FOR ARTIFICIAL SPEECH, invented by Sprague; and United States Patent No. 4,692,941, entitled REAL-TIME TEXT-TO-SPEECH CONVERSION SYSTEM, invented by Jacks, et al. Further background concerning speech synthesis may be found in United States Patent No. 4,384,169, entitled METHOD AND APPARATUS FOR SPEECH SYNTHESIZING, invented by Mozer, et al.

Notwithstanding the prior work in this area, the use of text-to-speech systems has not gained widespread acceptance. It is desirable therefore to provide a software only text-to-speech system which is portable to a wide variety of microcomputer platforms, and conserves memory space in such platforms for other uses.

The present invention is defined by the appended claims and provides a real time, text-to-speech system suitable for application in a wide variety of personal computer platforms which uses a relatively small amount of host system memory for execution.

According to the invention, there is provided an apparatus for synthesizing speech in response to a sequence of sound segment codes representing speech, comprising memory storing a set of quantization vectors having shaped quantization noise spectra, said quantization vectors being generated by an inverse noise shaping filter operation performed on a first set of quantization vectors that correspond to the sound segment codes;

means, responsive to sound segment codes in the sequence, for identifying strings of quantization vectors in the set of quantisation vectors having shaped quantisation noise spectra for respective sound segment codes in the sequence;

means, coupled to the means for identifying and the memory, for generating a speech data sequence in response to the strings of quantization vectors; and

an audio transducer, coupled to the means for generating, to generate sound in response to the speech data sequence.

The system is based on a speech compression algorithm which takes advantage of certain specialized knowledge concerning speech including the following:

1) Adjacent samples of the speech data are highly correlated. Thus a fixed linear prediction filter may be used to partially remove the correlation between adjacent samples.

2) In the case of voice to speech (e.g., vowels, nasals, etc.), the speech wave forms can be regarded as slowly varying periodic signals. Thus, an adaptive pitch predictor can be used to remove the redundancy in speech data and achieve a high data compression.

3) Finally, vector quantization is an extremely efficient approach to code correlated data vectors. It can be applied to partially de-correlated speech data according to the present invention, and noise shaping can be incorporated into the vector quantization process to improve the subjective quality of the synthesized speech. Further, a variety of different compression rates can be achieved by simply varying the vector size used for vector quantization.

Thus, according to one embodiment, the invention defined in claim 1 is an apparatus for synthesizing speech in response to a sequence of sound segment codes representing speech. The system includes a memory storing a set of noise compensated quantization vectors. A processing module in the apparatus is responsive to the sound segment codes in the sequence to identify strings of noise compensated quantization vectors in the set for respective sound segment codes in the sequence. A second processing module generates a speech data sequence in response to the strings of noise compensated quantization vectors. Finally, an audio transducer is coupled to the processing modules, and generates sound in response to the speech data sequence.

For noise compensation according to this aspect, sounds are encoded using noise shaped data and a first set of quantization vectors adapted for the noise shaped data. In decoding, a second set of noise compensated vectors different from the first set are used to recover improved quality sound.

Another embodiment of the invention, as claimed, additionally, involves utilizing the quantization vectors to represent filtered sound segment data, and providing for a module for applying an inverse filter to the strings of quantization vectors in the generation of the speech data sequence. According to this aspect, the quantization vectors may represent a quantization of results of linear prediction filtering of sound segment data for spectral flattening to de-correlate the sound samples used for quantization and the quantization noise. In decompressing the sound segment data, an inverse linear prediction filter is applied to the identified strings of quantization vectors to recover the sound data. Also, the quantization vectors represent quantization of results of pitch filtering of sound segment data. Thus, an inverse pitch filter is applied to the identified strings of quantization vectors in the module of generating the speech data sequence.

In systems using the inverse linear prediction filter and the inverse pitch filter, the sound segment codes also include parameters used in executing the inverse filtering steps. In the preferred system, these parameters are chosen, along with filter coefficients used in the decoding, so that the decoding can be executed without multiplication. That is, shifts and adds replace any multiplication required by these specifically chosen values.

According to another aspect of the invention, there is provided an apparatus for synthesizing speech in response to a text, comprising:

means for translating text to a sequence of sound segment codes;
means for generating a set of quantization vectors having shaped quantization noise spectra by applying a noise shaping filter function is to a first set of quantization vectors that correspond to the sound segment codes;
memory storing the set of quantization vectors having shaped quantization noise spectra;
means, responsive to sound segment codes in the sequence, for identifying strings of quantization vectors in the set of quantization vectors having shaped quantization noise spectra for respective sound segment codes in the sequence;
means, coupled to the means for identifying and the memory, for generating a speech data sequence in response to the strings of quantization vectors; and
an audio transducer, coupled to the means for generating, to generate sound in response to the speech data sequence.

According to a further aspect, there is provided an apparatus for synthesizing speech in response to a text, comprising:

a programmable processor to execute routines to produce a speech data sequence;
an audio transducer, coupled to the processor, to generate sound in response to the speech data sequence;
a table memory, coupled to the processor, storing a noise-shaped set of quantization vectors produced by performing an inverse noise shaping filter operation on a first set of quantization vectors, and a table of encoded diphones having entries including data identifying a string of quantization vectors in the said noise-shaped for respective diphones; and
an instruction memory, coupled to the processor, storing a translator routine for execution by the processor to translate the text to a sequence of diphone indices, and a decoder routine for execution by the processor including

means, responsive to diphone indices in the sequence, for accessing the table of encoded diphones to identify

strings of quantization vectors in the said noise-shaped set for diphones in the text; and
means, coupled to the means for accessing and the table memory, for retrieving the identified strings of quantization vectors;
means, coupled with the means for retrieving, for producing diphone data strings in response to the identified strings of quantization vectors, wherein the diphone data strings each have a beginning and an ending;
means, coupled to the means for producing, for blending the ending of a particular diphone data string in the sequence with the beginning of an adjacent diphone data string in the sequence to smooth discontinuities between the particular and adjacent diphone data strings to produce a smoothed string of quantized speech data; and
means, responsive to the text and the smoothed string of quantized speech data, for adjusting pitch and duration of the identified strings of quantization vectors for the diphones in the sequence to produce the speech data sequence for supply to the audio transducer.

The invention can therefore be defined according to any of appended claims 13-32 as an apparatus for synthesizing speech in response to text. This system includes a module that translates received text into a sequence of sound segments codes which are decoded as described above. The text translator includes a table of encoded diphones having entries that include data identifying a string of quantization vectors in the set for the respective diphones. The sequence of sound segment codes thus comprises a sequence of indices to the table of encoded diphones representing the text. The strings of the quantization vectors for a given sound segment code are identified by accessing the entries in the table of encoded diphones.

The module for generatihg the speech data waveform may also include modules for improving the quality of the synthesized speech. Such modules include a routine for blending the ending of a particular diphone in the sequence with beginning of an adjacent diphone to smooth discontinuities between the particular and adjacent diphone data strings. Further, the string of quantized speech data may be applied to a system which adjusts the pitch and duration of the sounds represented by the strings of quantization vectors.

According to yet another embodiment of the invention, the apparatus for synthesizing speech may additionnally include an encoder for generating the table of encoded diphones. In this aspect, the encoder receives sampled speech for the respective diphones, applies a fixed linear prediction filter to partially de-correlate the speech samples and the quantization noise, applies a pitch filter to the output of the linear prediction filter, and applies a noise shaping filter to generate a resulting set of vectors. The resulting set of vectors is then matched to vectors in a vector quantization table. The vectors in the vector quantization table are related to the quantization vectors used for decoding the speech data by the same noise shaping filter or a derivative of it to subjectively improve the quality of the decompressed speech.

This encoding technique allows use of the decoding technique which is very simple, requires a small amount of memory, and produces very high quality speech.

In the text-to-speech system a higher level of compression is achieved while keeping the decoder complexity to an absolute minimum. The compression ratio can be varied depending on the available RAM in the computer. In order to store speech in an uncompressed form, normally 8-16 bits per sample is required. The number of bits required to store each sample can be reduced to 0.5 bits (i.e., about 16 samples of speech can be stored using 8 bits of memory). However, higher quality synthesized speech can be produced when larger RAM space is available, using about 4 bits per sample. A speech compression/decompression technique is also described.

Other aspects and advantages of the present invention can be seen from the detailed description of preferred embodiments, given by way of example only, taken in conjunction with the drawings, and the claims which follow.

Fig. 1 is a block diagram of a generic hardware platform incorporating a text-to-speech system according to the present invention.

Fig. 2 is a flow chart illustrating a basic text-to-speech routine according to the present invention.

Fig. 3 illustrates the format of diphone records according to one embodiment of the present invention.

Fig. 4 is a flow chart illustrating an encoder for speech data for use with the present invention.

Fig. 5 is a graph discussed in reference to the estimation of pitch filter parameters in the encoder of Fig. 4.

Fig. 6 is a flow chart illustrating the full search used in the encoder of Fig. 4.

Fig. 7 is a flow chart illustrating a decoder for speech data according to the present invention.

Fig. 8 is a flow chart illustrating a technique for blending the beginning and ending of adjacent diphone records.

Fig. 9 consists of a set of graphs referred to in explanation of the blending technique of Fig. 8.

Fig. 10 is a graph illustrating a typical pitch versus time diagram for a sequence of frames of speech data.

Fig. 11 is a flow chart illustrating a technique for increasing the pitch period of a particular frame.

Fig. 12 is a set of graphs referred to in explanation of the technique of Fig. 11.

Fig. 13 is a flow chart illustrating a technique for decreasing the pitch period of a particular frame.

Fig. 14 is a set of graphs referred to in explanation of the technique of Fig. 13.

Fig. 15 is a flow chart illustrating a technique for inserting a pitch period between two frames in a sequence.

Fig. 16 is a set of graphs referred to in explanation of the technique of Fig. 15.

Fig. 17 is a flow chart illustrating a techniqu for deleting a pitch period in a sequence of frames.

Fig. 18 is a set of graphs referred to in explanation of the technique of Fig. 17.

A detailed description of preferred embodiments of the present invention is provided with reference to the figures. Figs. 1 and 2 provide a overview of a system incorporating the present invention. Fig. 3 illustrates the basic manner in which diphone records are stored according to the present invention. Figs. 4-6 illustrate encoding methods based on vector quantization. Fig. 7 illustrates the decoding algorithm according to the present invention.

Figs. 8 and 9 illustrate a preferred technique for blending the beginning and ending of adjacent diphone records. Figs. 10-18 illustrate the techniques for controlling the pitch and duration of sounds in the text-to-speech system.

1. System Overview (Figs. 1-3)

Fig. 1 illustrates a basic microcomputer platform incorporating a text-to-speech system based on vector quantization according to the present invention. The platform includes a central processing unit 10 coupled to a host system bus 11. A keyboard 12 or other text input device is provided in the system. Also, a display system 13 is coupled to the host system bus. The host system also includes a non-volatile storage system such as a disk drive 14. Further, the system includes host memory 15. The host memory includes text-to-speech (TTS) code, including encoded voice tables, buffers, and other host memory. The text-to-speech code is used to generate speech data for supply to an audio output module 16 which includes a speaker 17.

According to the present invention, the encoded voice tables include a TTS dictionary which is used to translate text to a string of diphones. Also included is a diphone table which translates the diphones to identified strings of quantization vectors. A quantization vector table is used for decoding the sound segment codes of the diphone table into the speech data for audio output. Also, the system may include a vector quantization table for encoding which is loaded into the host memory 15 when necessary.

The platform illustrated in Fig. 1 represents any generic microcomputer system, including a Macintosh based system, a DOS based system, a UNIX based system or other types of microcomputers. The text-to-speech code and encoded voice tables according to the present invention for decoding occupy a relatively small amount of host memory 15. For instance, a text-to-speech decoding system according to the present invention may be implemented which occupies less than 640 kilobytes of main memory, and yet produces high quality, natural sounding synthesized speech.

The basic algorithm executed by the text-to-speech code is illustrated in Fig. 2. The system first receives the input text (block 20). The input text is translated to diphone strings using the TTS dictionary (block 21). At the same time, the input text is analyzed to generate intonation control data, to control the pitch and duration of the diphones making up the speech (block 22).

After the text has been translated to diphone strings, the diphone strings are decompressed to generate vector quantized data frames (block 23). After the vector quantized (VQ) data frames are produced, the beginnings and endings of adjacent diphones are blended to smooth any discontinuities (block 24). Next, the duration and pitch of the diphone VQ data frames are adjusted in response to the intonation control data (block 25 and 26). Finally, the speech data is supplied to the audio output system for real time speech production (block 2-7). For systems having sufficient processing power, an adaptive post filter may be applied to further improve the speech quality.

The TTS dictionary can be implemented using any one of a variety of techniques known in the art. According to the present invention, diphone records are implemented as shown in Fig. 3 in a highly compressed format.

As shown in Fig. 3, records for a left diphone 30 and a right diphone 31 are shown. The record for the left diphone 30 includes a count 32 of the number NL of pitch periods in the diphone. Next, a pointer 33 is included which points to a table of length NL storing the number LP_i for each pitch period, i goes from 0 to NL-1 of pitch values for corresponding compressed frame records. Finally, pointer 34 is included to point to a table 36 of ML vector quantized compressed speech records, each having a fixed set length of encoded frame size related to nominal pitch of the encoded speech for the left diphone. The nominal pitch is based upon the average number of samples for a given pitch period for the speech data base.

A similar structure can be seen for the right diphone 31. Using vector quantization, a length of the compressed speech records is very short relative to the quality of the speech generated.

The format of the vector quantized speech records can be understood further with reference to the frame encoder routine and the frame decoder routine described below with reference to Figs. 4-7.

II. The Encoder/Decoder Routines (Figs. 4-7)

The encoder routine is illustrated in Fig. 4. The encoder accepts as input a frame s_n of speech data. In the preferred system, the speech samples are represented as 12 or 16 bit two's complement numbers, sampled at 22,252 Hz. This data is divided into non-overlapping frames s_n having a length of N, where N is referred to as the frame size. The value

of N depends on the nominal pitch of the speech data. If the nominal pitch of the recorded speech is less than 165 samples (or 135 Hz), the value of N is chosen to be 96. Otherwise a frame size of 160 is used. The encoder transforms the N-point data sequence s_n into a byte stream of shorter length, which depends on the desired compression rate. For example, if N=160 and very high data compression is desired, the output byte stream can be as short as 12 eight bit bytes. A block diagram of the encoder is shown in Fig. 4.

Thus, the routine begins by accepting a frame s_n (block 50). To remove low frequency noise, such as DC or 60 Hz power line noise, and produce offset free speech data, signal s_n is passed through a high pass filter. A difference equation used in a preferred system to accomplish this is set out in Equation 1 for $0 \leq n < N$.

$$x_n = s_n - s_{n-1} + 0.999 * x_{n-1} \qquad \text{Equation 1}$$

The value x_n is the "offset free" signal. The variables s_{-1} and x_{-1} are initialized to zero for each diphone and are subsequently updated using the relation of Equation 2.

$$x_{-1} = x_N \text{ and } s_{-1} = s_N \qquad \text{Equation 2}$$

This step can be referred to as offset compensation or DC removal (block 51).

In order to partially decorrelate the speech samples and the quantization noise, the sequence x_n is passed through a fixed first order linear prediction filter. The difference equation to accomplish this is set forth in Equation 3.

$$y_n = x_n - 0.875 * x_{n-1} \qquad \text{Equation 3}$$

The linear prediction filtering of Equation 3 produces a frame y_n (block 52). The filter parameter, which is equal to 0.875 in Equation 3, will have to be modified if a different speech sampling rate is used. The value of x_{-1} is initialized to zero for each diphone, but will be updated in the step of inverse linear prediction filtering (block 60) as described below.

It is possible to use a variety of filter types, including, for instance, an adaptive filter in which the filter parameters are dependent on the diphones to be encoded, or higher order filters.

The sequence y_n produced by Equation 3 is then utilized to determine an optimum pitch value, P_{opt}, and an associated gain factor, β. P_{opt} is computed using the functions $s_{xy}(P)$, $s_{xx}(P)$, $s_{yy}(P)$, and the coherence function Coh(P) defined by Equations 4, 5, 6 and 7 as set out below.

$$s_{xy}(P) = \sum_{n=0}^{N-1} y_n * PBUF_{P_{max} - P + n}$$

Equation 4

$$s_{xx}(P) = \sum_{n=0}^{N-1} y_n * y_n$$

Equation 5

$$s_{yy}(P) = \sum_{N=0}^{N-1} PBUF_{P_{max} - P + n} * PBUF_{P_{max} - P + n} \qquad \text{Equation 6}$$

and

$$Coh(P) = s_{xy}(P) * s_{xy}(P) / (s_{xx}(P) * s_{yy}(P)) \qquad \text{Equation 7}$$

PBUF is a pitch buffer of size P_{max}, which is initialized to zero, and updated in the pitch buffer update block 59 as described below. P_{opt} is the value of P for which Coh(P) is maximum and $s_{xy}(P)$ is positive. The range of P considered depends on the nominal pitch of the speech being coded. The range is (96 to 350) if the frame size is equal to 96 and is (160 to 414) if the frame size is equal to 160. P_{max} is 350 if nominal pitch is less than 160 and is equal to 414 otherwise. The parameter P_{opt} can be represented using 8 bits.

The computation of P_{opt} can be understood with reference to Fig. 5. In Fig. 5, the buffer PBUF is represented by the sequence 100 and the frame y_n is represented by the sequence 101. In a segment of speech data in which the preceding frames are substantially equal to the frame y_n, PBUF and y_n will look as shown in Fig. 5. P_{opt} will have the value at point 102, where the vector y_n 101 matches as closely as possible a corresponding segment of similar length in PBUF 100.

The pitch filter gain parameter β is determined using the expression of Equation 8.

$$\beta = s_{xy}(P_{opt}) / s_{yy}(P_{opt}). \qquad \text{Equation 8}$$

β is quantized to four bits, so that the quantized value of β can range from 1/16 to 1, in steps of 1/16.

Next, a pitch filter is applied (block 54). The long term correlations in the pre-emphasized speech data y_n are removed using the relation of Equation 9.

$$r_n = y_n - \beta * PBUF_{P_{max} - P_{opt} + n}, \qquad 0 \le n < N. \qquad \text{Equation 9}$$

This results in computation of a residual signal r_n.

Next, a scaling parameter G is generated using a block gain estimation routine (block 55). In order to increase the computational accuracy of the following stages of processing, the residual signal r_n is rescaled. The scaling parameter, G, is obtained by first determining the largest magnitude of the signal r_n and quantizing it using a 7-level quantizer. The parameter G can take one of the following 7 values: 256, 512, 1024, 2048, 4096, 8192, and 16384. The consequence of choosing these quantization levels is that the rescaling operation can be implemented using only shift operations.

Next the routine proceeds to residual coding using a full search vector quantization code (block 56). In order to code the residual signal r_n, the n point sequence r_n is divided into non-overlapping blocks of length M, where M is referred to as the "vector size". Thus, M sample blocks b_{ij} are created, where i is an index from zero to M- on the block number, and j is an index from zero to N/M-1 on the sample within the block. Each block may be defined as set out in Equation 10.

$$b_{ij} = r_{Mi+j}, \ (0 \le i < N/M \text{ and } j \le 0 < M) \qquad \text{Equation 10}$$

Each of these M sample blocks b_{ij} will be coded into an 8 bit number using vector quantization. The value of M depends on the desired compression ratio. For example, with M equal to 16, very high compression is achieved (i.e., 16 residual samples are coded using only 8 bits). However, the decoded speech quality can be perceived to be somewhat noisy with M=16. On the other hand, with M=2, the decompressed speech quality will be very close to that of uncompressed speech. However the length of the compressed speech records will be longer. In the preferred inplementation, the value M can take values 2, 4, 8 and 16.

The vector quantization is performed as shown in Fig. 6. Thus, for all blocks b_{ij} a sequence of quantization vectors is identified (block 120). First, the components of block b_{ij} are passed through a noise shaping filter and scaled as set out in Equation 11 (block 121).

$$w_j = 0.875 * w_{j-1} - 0.5 * w_{j-2} + 0.4375 * w_{j-3} + b_{ij}, \qquad 0 \le j < M$$

$$v_{ij} = G * w_j \qquad 0 \le j < M \qquad \text{Equation 11}$$

Thus, v_{ij} is the jth component of the vector v_i, and the values w_{-1}, w_{-2} and w_{-3} are the states of the noise shaping filter and are initialized to zero for each diphone. The filter coefficients are chosen to shape the quantization noise spectra in order to improve the subjective quality of the decompressed speech. After each vector is coded and decoded, these states are updated as described below with reference to blocks 124-126.

Next, the routine finds a pointer to the best match in a vector quantization table (block 122). The vector quantization table 123 consists of a sequence of vectors C_0 through C_{255} (block 123).

Thus, the vector v_i is compared against 256 M-point vectors, which are precomputed and stored in the code table 123. The vector C_{qi} which is closest to v_i is determined according to Equation 12. The value C_p for p =0 through 255 represents the p^{th} encoding vector from the vector quantization code table 123.

$$\min_{p} \sum_{j=0}^{M-1} (v_{ij} - C_{pj})^2$$

Equation 12

The closest vector C_{qi} can also be determined efficiently using the technique of Equation 13.

$$v_i^T \bullet C_{qi} \le v_i^T \bullet C_p \text{ for all } p(0 \le p \le 255) \qquad \text{Equation 13}$$

In Equation 13, the value v^T represents the transpose of the vector v, and "•" represents the inner product operation in the inequality.

The encoding vectors C_p in table 123 are utilized to match on the noise filtered value v_{ij}. However in decoding, a decoding vector table 125 is used which consists of a sequence of vectors QV_p. The values QV_p are selected for the purpose of achieving quality sound data using the vector quantization technique. Thus, after finding the vector C_{qi}, the pointer q is utilized to access the vector QV_{qi}. The decoded samples corresponding to the vector b_i which is produced at step 55 of Fig. 4, is the M-point vector (1/G) * QV_{qi}. The vector C_p is related to the vector QV_p by the noise shaping filter operation of Equation 11. Thus, when the decoding vector QV_p is accessed, no inverse noise shaping filter needs to be computed in the decode operation. The table 125 of Fig. 6 thus includes noise compensated quantization vectors.

In continuing to compute the encoding vectors for the vectors b_{ij} which make up the residual signal r_n, the decoding vector of the pointer to the vector b_i is accessed (block 124). That decoding vector is used for filter and PBUF updates (block 126).

For the noise shaping filter, after the decoded samples are computed for each sub-block b_i, the error vector (b_i-QV_{qi}) is passed through the noise shaping filter as shown in Equation 14.

$$W_j = 0.875 * W_{j-1} - 0.5 + W_{j-2} + 0.4375 * W_{j-3} + [b_{ij} - QV_{qi}(j)]$$

$$0 \le j < M \qquad \text{Equation 14}$$

In Equation 14, the value $QV_{qi}(j)$ represents the j^{th} component of the decoding vector QV_{qi}. The noise shaping filter states for the next block are updated as shown in Equation 15.

$$w_{-1} = w_{M-1}$$

$$w_{-2} = w_{M-2}$$

$$w_{-3} = w_{M-3} \qquad \text{Equation 15}$$

This coding and decoding is performed for all of the N/M subblocks to obtain N/M indices to the decoding vector table 125. This string of indices Q_n, for n going from zero to N/M-1 represent identifiers for a string of decoding vectors for the residual signal r_n.

Thus, four parameters represent the N-point data sequence y_n:

1)) Optimum pitch, P_{opt} (8 bits),
2) Pitch filter gain, β (4 bits),
3) Scaling parameter, G (3 bits), and
4) A string of decoding table indices, Q_n ($0 \le n < N/M$).

The parameters β and G can be coded into a single byte. Thus, only (N/M) plus 2 bytes are used to represent N samples of speech. For example, suppose nominal pitch is 100 samples long, and M=16. In this case, a frame of 96 samples of speech are represented by 8 bytes: 1 byte for P_{opt}, 1 byte for β and G, and 6 bytes for the decoding table indices Q_n. If the uncompressed speech consists of 16 bit samples, then this represents a compression of 24:1.

Back to Fig. 4, four parameters identifying the speech data are stored (block 57). In a preferred system, they are stored in a structure as described with respect to Fig. 3 where the structure of the frame can be characterized as follows:

```
#define     NumOfVectorsPerFrame  (FrameSize / VectorSize)

struct frame {
        unsigned    Gain : 4;
        unsigned    Beta : 3;
        unsigned    UnusedBit: 1;
        unsigned    char Pitch ;
        unsigned    char VQcodes[NumOfVectorsPerFrame]; };
```

The diphone record of Fig. 3 utilizing this frame structure can be characterized as follows:

```
DiphoneRecord
{
        char   LeftPhone, RightPhone;
        short  LeftPitchPeriodCount,RightPitchPeriodCount;
        short  *LeftPeriods, *RightPeriods;
        struct      frame *LeftData, *RightData;
}
```

These stored parameters uniquely provide for identification of the diphones required for text-to-speech synthesis.

As mentioned above with respect to Fig. 6, the encoder continues decoding the data being encoded in order to update the filter and PBUF values. The first step involved in this is an inverse pitch filter (block 58). With the vector r'_n corresponding to the decoded signal formed by concatenating the string of decoding vectors to represent the residual signal r'_n, the inverse filter is implemented as set out in Equation 16.

$$y'_n = r'_n + \beta * PBUF_{Pmax - Popt + n} \qquad 0 \le n < N. \qquad \text{Equation 16}$$

Next, the pitch buffer is updated (block 59) with the output of the inverse pitch filter. The pitch buffer PBUF is updated as set out in Equation 17.

$$PBUF_n = PBUF_{(n+N)} \qquad 0 \le n < (P_{max} - N)$$

$$PBUF_{(Pmax-N+n)} = y'_n \qquad 0 \le n < N \qquad \text{Equation 17}$$

Finally, the linear prediction filter parameters are updated using an inverse linear prediction filter step (block 60). The output of the inverse pitch filter is passed through a first order inverse linear prediction filter to obtain the decoded speech. The difference equation to implement this filter is set out in Equation 18.

$$x'_n = 0.875 * x'_{n-1} + y'_n \qquad \text{Equation 18}$$

In Equation 18, x'_n is the decompressed speech. From this, the value of x_{-1} for the next frame is set to the value x_N for use in the step of block 52.

Fig. 7 illustrates the decoder routine. The decoder module accepts as input (N/M) + 2 bytes of data, generated by the encoder module, and applies as output N samples of speech. The value of N depends on the nominal pitch of the speech data and the value of M depends on the desired compression ratio.

In software only text-to-speech systems, the computational complexity of the decoder must be as small as possible to ensure that the text-to-speech system can run in real time even on slow computers. A block diagram of the decoder is shown in Fig. 7.

The routine starts by accepting diphone records at block 200. The first step involves parsing the parameters G, β, P_{opt}, and the vector quantization string Q_n (block 201). Next, the residual signal r'_n is decoded (block 202). This involves accessing and concatenating the decoding vectors for the vector quantization string as shown schematically at block 203 with access to the decoding quantization vector table 125.

After the residual signal r'_n is decoded, an inverse pitch filter is applied (block 204). This inverse pitch filter is implemented as shown in Equation 19:

$$y'_n = r_n + \beta * SPBUF(P_{max} - P_{opt} + n), \qquad 0 \le n < N. \qquad \text{Equation 19}$$

SPBUF is a synthesizer pitch buffer of length P_{max} initialized as zero for each diphone, as described above with respect to the encoder pitch buffer PBUF.

For each frame, the synthesis pitch buffer is updated (block 205). The manner in which it is updated is shown in Equation 20:

$$SPBUF_n = SPBUF_{(n+N)} \qquad 0 \le n < P_{max} - N)$$

$$SPBUF_{(Pmax-N+n)} = y'_n \qquad 0 \le n < N \qquad \text{Equation 20}$$

After updating SPBUF, the sequence y'_n is applied to an inverse linear prediction filtering step (block 206). Thus, the output of the inverse pitch filter y'_n is passed through a first order inverse linear prediction filter to obtain the decoded speech. The difference equation to implement the inverse linear prediction filter is set out in Equation 21:

$$x'_n = 0.875 * x'_{n-1} + y'_n \qquad \text{Equation 21}$$

In Equation 21, the vector x'_n corresponds to the decompressed speech. This filtering operation can be implemented using simple shift operations without requiring any multiplication. Therefore, it executes very quickly and utilizes a very small amount of the host computer resources.

Encoding and decoding speech according to the algorithms described above, provide several advantages over prior art systems. First, this technique offers higher speech compression rates with decoders simple enough to be used in the implementation of software only text-to-speech systems on computer systems with low processing power. Second, the technique offers a very flexible trade-off between the compression ratio and synthesizer speech quality. A high-end computer system can opt for higher quality synthesized speech at the expense of a bigger RAM memory requirement.

III. Waveform Blending For Discontinuity Smoothing (Figs. 8 and 9)

As mentioned above with respect to Fig. 2, the synthesized frames of speech data generated using the vector quantization technique may result in slight discontinuities between diphones in a text string. Thus, the text-to-speech system provides a module for blending the diphone data frames to smooth such discontinuities. The blending technique of the preferred embodiment is shown with respect to Figs. 8 and 9.

Two concatenated diphones will have an ending frame and a beginning frame. The ending frame of the left diphone must be blended with the beginning frame of the right diphone without audible discontinuities or clicks being generated. Since the right boundary of the first diphone and the left boundary of the second diphone correspond to the same phoneme in most situations, they are expected to be similar looking at the point of concatenation. However, because the two diphone codings are extracted from different context, they will not look identical. This blending technique is applied to eliminate discontinuities at the point of concatenation. In Fig. 9, the last frame, referring here to one pitch period, of the left diphone is designated L_n ($0 \le n < PL$) at the top of the page. The first frame (pitch period) of the right diphone is designated R_n ($0 \le n < PR$). The blending of L_n and R_n according to the present invention will alter these two pitch periods only and is performed as discussed with reference to Fig. 8. The waveforms in Fig. 9 are chosen to illustrate the algorithm, and may not be representative of real speech data.

Thus, the algorithm as shown in Fig. 8 begins with receiving the left and right diphone in a sequence (block 300). Next, the last frame of the left diphone is stored in the buffer L_n (block 301). Also, the first frame of the right diphone is stored in buffer R_n (block 302).

Next, the algorithm replicates and concatenates the left frame L_n to form extend frame (block 303). In the next step, the discontinuities in the extended frame between the replicated left frames are smoothed (block 304). This smoothed and extended left frame is referred to as EI_n in Fig. 9.

The extended sequence EI_n ($0 \le n < PL$) is obtained in the first step as shown in Equation 22:

$$EI_n = L_n \qquad n = 0,1,\ldots,PL-1$$

$$EI_{PL+n} = L_n \qquad n = 0,1,\ldots,PL-1 \qquad \text{Equation 22}$$

Then discontinuity smoothing from the point $n = P^L$ is conducted according to the filter of Equation 23:

$$EI_{PL+n} = EI_{PL+n} + [EI_{(PL-1)} - EI'_{(PL-1)}]^*\Delta^{n+1},$$

$$n = 0,1,\ldots,(PL/2). \qquad \text{Equation 23}$$

In Equation 23, the value Δ is equal to 15/16 and $EI'_{(PL-1)} = EI_2 + 3 * (EI_1 - EI_0)$. Thus, as indicated in Fig. 9, the extended sequence EI_n is substantially equal to L_n on the left hand side, has a smoothed region beginning at the point P_L and converges on the original shape of L_n toward the point $2P_L$. If L_n was perfectly periodic, then $EI_{PL-1} = EI'_{PL-1}$.

In the next step, the optimum match of R_n with the vector EI_n is found. This match point is referred to as P_{opt}. (Block 305.) This is accomplished essentially as shown in Fig. 9 by comparing R_n with EI_n to find the section of EI_n which most closely matches R_n. This optimum blend point determination is performed using Equation 23 where W is the minimum of PL and PR, and AMDF represents the average magnitude difference function.

$$AMDF(p) = \sum_{n=0}^{W-1} | EI_{n+p} - R_n |$$

Equation 24

This function is computed for values of p in the range of 0 to PL-1. The vertical bars in the operation denote the absolute value. W is the window size for the AMDF computation. P_{opt} is chosen to be the value at which AMDF(p) is minimum. This means that $p = P_{opt}$ corresponds to the point at which sequences EI_{n+p} ($0 \le n < W$) and R_n ($0 \le n < W$) are very close to each other.

After determining the optimum blend point P_{opt}, the waveforms are blended (block 306). The blending utilizes a

first weighting ramp WL which is shown in Fig. 9 beginning at P_{opt} in the EI_n trace. In a second ramp, WR is shown in Fig. 9 at the R_n trace which is lined up with P_{opt}. Thus, in the beginning of the blending operation, the value of EI_n is emphasized. At the end of the blending operation, the value of R_n is emphasized.

Before blending, th length PL of L_n is altered as needed to ensure that when the modified L_n and R_n are concatenated, the waveforms are as continuous as possible. Thus, the length P'L is set to P_{opt} if P_{opt} is greater than PL/2. Otherwise, the length P'L is equal to W + P_{opt} and the sequence L_n is equal to EI_n for $0 \le n \le (P'L-1)$.

The blending ramp beginning at P_{opt} is set out in Equation 25:

$$R_n = EI_{n+Popt} + (R_n - EI_{n+Popt})^*(n+1)/W \qquad 0 \le n < W$$

$$R = R_n \qquad W \le n < PR \qquad \text{Equation 25}$$

Thus, the sequences L_n and R_n are windowed and added to get the blended R_n. The beginning of L_n and the ending of R_n are preserved to prevent any discontinuities with adjacent frames.

This blending technique is believed to minimize blending noise in synthesized speech produced by any concatenated speech synthesis.

IV. Pitch and Duration Modification (Figs. 10-18)

As mentioned above with respect to Fig. 2, a text analysis program analyzes the text and determines the duration and pitch contour of each phone that needs to be synthesized and generates intonation control signals. A typical control for a phone will indicate that a given phoneme, such as AE, should have a duration of 200 milliseconds and a pitch should rise linearly from 220Hz to 300Hz. This requirement is graphically shown in Fig. 10. As shown in Fig. 10, T equals the desired duration (e.g. 200 milliseconds) of the phoneme. The frequency f_b is the desired beginning pitch in Hz. The frequency f_e is the desired ending pitch in Hz. The labels P_1, P_2...,P_6 indicate the number of samples of each frame to achieve the desired pitch frequencies f_b, f_2...,f_6. The relationship between the desired number of samples, P_i, and the desired pitch frequency f_i ($f_1 = f_b$), is defined by the relation:

$P_i = F_s/f_i$, where F_s is the sampling frequency for the data. As can be seen in Fig. 10, the pitch period for a lower frequency period of the phoneme is longer than the pitch period for a higher frequency period of the phoneme. If the nominal frequency were P_3, then the algorithm would be required to lengthen the pitch period for frames P_1 and P_2 and decrease the pitch periods for frames P_4, P_5 and P_6. Also, the given duration T of the phoneme will indicate how many pitch periods should be inserted or deleted from the encoded phoneme to achieve the desired duration period. Figs. 11 through 18 illustrate a preferred implementation of such algorithms.

Fig. 11 illustrates an algorithm for increasing the pitch period, with reference to the graphs of Fig. 12. The algorithm begins by receiving a control to increase the pitch period to N + Δ, where N is the pitch period of the encoded frame. (Block 350). In the next step, the pitch period data is stored in a buffer x_n (block 351). x_n is shown in Fig. 12 at the top of the page. In the next step, a left vector L_n is generated by applying a weighting function WL to the pitch period data x_n with reference to Δ (block 352). This weighting function is illustrated in Equation 26 where M = N-Δ:

$$L_n = x_n \qquad \text{for } 0 \le n < \Delta$$

$$L_n = x_n * (N-n)/(M+1) \qquad \text{for } \Delta \le n < N \qquad \text{Equation 26}$$

As can be seen in Fig. 12, the weighting function WL is constant from the first sample to sample Δ, and decreases from Δ to N.

Next, a weighting function WR is applied to x_n (block 353) as can be seen in the Fig. 12. This weighting function is executed as shown in Equation 27:

$$R_n = x_{n+\Delta} * (n+1)/(M+1) \qquad \text{for } 0 \le n < N-\Delta$$

$$R_n = x_{n+\Delta} \qquad \text{for } N-\Delta \le n < N. \qquad \text{Equation 27}$$

As can be seen in Fig. 12, the weighting function WR increases from 0 to N-Δ and remains constant from N-Δ to N. The resulting waveforms L_n and R_n are shown conceptually in Fig. 12. As can be seen, L_n maintains the beginning

of the sequence x_n, while R_n maintains the ending of the data x_n.

The pitch modified sequence y_n is formed (block 354) by adding the two sequences as shown in Equation 28:

$$y_n = L_n + R_{(n-\Delta)} \qquad \text{Equation 28}$$

This is graphically shown in Fig. 12 by placing R_n shifted by Δ below L_n. The combination of L_n and R_n shifted by Δ is shown to be y_n at the bottom of Fig. 12. The pitch period for y_n is $N + \Delta$. The beginning of y_n is the same as the beginning of x_n, and the ending of y_n is substantially the same as the ending of x_n. This maintains continuity with adjacent frames in the sequence, and accomplishes a smooth transition while extending the pitch period of the data.

Equation 28 is executed with the assumption that L_n is 0, for $n \leq N$, and R_n is 0 for $n<0$. This is illustrated pictorially in Fig. 12.

An efficient implementation of this scheme which requires at most one multiply per sample, is shown in Equation 29:

$$y_n = x_n \qquad 0 \leq n < \Delta$$

$$y_n = x_n + [x_{n-\Delta} - x_n] * (n-\Delta + 1)/(N-\Delta + 1) \qquad \Delta \leq n < N$$

$$y_n = x_{n-\Delta} \qquad N \leq n < N_d \qquad \text{Equation 29}$$

This results in a new pitch period having a pitch period of $N + \Delta$.

There are also instances in which the pitch period must be decreased. The algorithm for decreasing the pitch period is shown in Fig. 13 with reference to the graphs of Fig. 14. Thus, the algorithm begins with a control signal indicating that the pitch period must be decreased to $N-\Delta$. (Block 400). The first step is to store two consecutive pitch periods in the buffer x_n (block 401). Thus, the buffer x_n as can be seen in Fig. 14 consists of two consecutive pitch periods, with the period N_l being the length of the first pitch period, and N_r being the length of the second pitch period. Next, two sequences L_n and R_n are conceptually created using weighting functions WL and WR (blocks 402 and 403). The weighting function WL emphasizes the beginning of the first pitch period, and the weighting function WR emphasizes the ending of the second pitch period. These functions can be conceptually represented as shown in Equations 30 and 31, respectively:

$$L_n = x_n \qquad \text{for } 0 \leq n < N_l - W$$

$$L_n = x_n * (N_l-n)/(W+ 1) \qquad W \leq n < N_l$$

$$L_n = 0 \qquad \text{otherwise.} \qquad \text{Equation 30}$$

and

$$R_n = x_n * (n-N_l+W-\Delta+1)/(W+ 1) \qquad \text{for } N_l - W + \Delta \leq n < N_l + \Delta$$

$$R_n = x_n \qquad \text{for } N_l + \Delta \leq n < N_l + N_r$$

$$R_n = 0 \qquad \text{otherwise.} \qquad \text{Equation 31}$$

In these equations, Δ is equal to the difference between N_l and the desired pitch period N_d. The value W is equal to $2*\Delta$, unless $2*\Delta$ is greater than N_d, in which case W is equal to N_d.

These two sequences L_n and R_n are blended to form a pitch modified sequence y_n (block 404). The length of the pitch modified sequence y_n will be equal to the sum of the desired length and the length of the right phoneme frame N_r. It is formed by adding the two sequences as shown in Equation 32:

$$y_n = L_n + R_{(n + \Delta)} \qquad \text{Equation 32}$$

Thus, when a pitch period is decreased, two consecutive pitch periods of data are affected, even though only the length of one pitch period is changed. This is done because pitch periods are divided at places where short-term energy is the lowest within a pitch period. Thus, this strategy affects only the low energy portion of the pitch periods. This minimizes the degradation in speech quality due to the pitch modification. It should be appreciated that the drawings in Fig. 14 are simplified and do not represent actual pitch period data.

An efficient implementation of this scheme, which requires at most one multiply per sample, is set out in Equations 33 and 34.

The first pitch period of length N_d is given by Equation 33:

$$y_n = x_n \qquad 0 \le n < N_l - W$$

$$y_n = x_n + [x_{n+\Delta} - x_n]*(n-N_l + W + 1)/(W + 1) \qquad N_l - W \le n < N_d \qquad \text{Equation 33}$$

The second pitch period of length N_r is generated as shown in Equation 34:

$$y_n = x_{n-\Delta} + [x_n - x_{n-\Delta}]*(n-\Delta-N_l + W + 1)/(W +1) \qquad N_l \le n < N_l + \Delta$$

$$y_n = x_n \qquad N_{l+\Delta} \le n < N_l + N_r \qquad \text{Equation 34}$$

As can be seen in Fig. 14, the sequence L_n is essentially equal to the first pitch period until the point N_l-W. At that point, a decreasing ramp WL is applied to the signal to dampen the effect of the first pitch period.

As also can be seen, the weighting function WR begins at the point N_l-W + Δ and applies an increasing ramp to the sequence x_n until the point N_l + Δ From that point, a constant value is applied. This has the effect of damping the effect of the right sequence and emphasizing the left during the beginning of the weighting functions, and generating an ending segment which is substantially equal to the ending segment of x_n emphasizing the right sequence and damping the left. When the two functions are blended, the resulting waveform y_n is substantially equal to the beginning of x_n at the beginning of the sequence. At the point N_l-W a modified sequence is generated until the point N_l. From N_l to the ending, sequence x_n shifted by Δ results.

A need also arises for insertion of pitch periods to increase the duration of a given sound. A pitch period is inserted according to the algorithm shown in Fig. 15 with reference to the drawings of Fig. 16.

The algorithm begins by receiving a control signal to insert a pitch period between frames L_n and R_n (block 450). Next, both L_n and R_n are stored in the buffer (block 451), where L_n and R_n are two adjacent pitch periods of a voice diphone. (Without loss of generality, it is assumed for the description that the two sequences are of equal lengths N.)

In order to insert a pitch period, x_n of the same duration, without causing a discontinuity between L_n and x_n and between x_n and R_n, the pitch period x_n should resemble R_n around n =0 (preserving L_n to x_n continuity), and should resemble L_n around n = N (preserving x_n to R_n continuity). This is accomplished by defining x_n as shown in Equation 35:

$$x_n = R_n + (L_n - R_n) * [(n + 1)/(N + 1)] \qquad 0 \le n < N\text{-}1 \qquad \text{Equation 35}$$

Conceptually, as shown in Fig. 15, the algorithm proceeds by generating a left vector WL(L_n), essentially applying to the increasing ramp WL to the signal L_n. (Block 452).

A right vector WR (R_n) is generated using the weighting vector WR (block 453) which is essentially a decreasing ramp as shown in Fig. 16. Thus, the ending of L_n is emphasized with the left vector, and the beginning of R_n is emphasized with the vector WR.

Next, WR (L_n) and WR (R_n) are blended to create an inserted period x_n (block 454).

The computation requirement for inserting a pitch period is thus just a multiplication and two additions per speech sample.

Finally, concatenation of L_n, x_n and R_n produces a sequence with an inserted pitch period (block 455).

Deletion of a pitch period is accomplished as shown in Fig. 17 with reference to the graphs of Fig. 18. This algorithm, which is very similar to the algorithm for inserting a pitch period, begins with receiving a control signal indicating deletion of pitch period R_n which follows L_n (block 500). Next, the pitch periods L_n and R_n are stored in the buffer (block 501). This is pictorially illustrated in Fig. 18 at the top of the page. Again, without loss of generality, it is assumed that the two sequences have equal lengths N.

The algorithm operates to modify the pitch period L_n which precedes R_n (to be deleted) so that it resembles R_n,

as n approaches N. This is done as set forth in Equation 36:

$$L'_n = L_n + (R_n - L_n) * [(n+1)/(N+1)] \qquad 0 \le n < N-1 \qquad \text{Equation 36}$$

In Equation 36, the resulting sequence L'_n is shown at the bottom of Fig. 18. Conceptually, Equation 36 applies a weighting function WL to the sequence L_n (block 502). This emphasizes the beginning of the sequence L_n as shown. Next, a right vector WR (R_n) is generated by applying a weighting vector WR to the sequence R_n that emphasizes the ending of R_n (block 503).

WL (L_n) and WR (R_n) are blended to create the resulting vector L'_n. (Block 504). Finally, the sequence L_n-R_n is replaced with the sequence L'_n in the pitch period string. (Block 505).

IV. Conclusion

Accordingly, the present invention presents a text-to-speech system or system for translating sound segment codes representing speech, to speech, which is efficient, uses a very small amont of memory, and is portable to a wide variety of standard microcomputer platforms. It takes advantage of knowledge about speech data, and to create a speech compression, blending, and duration control routine which produces very high quality speech with very little computational resources.

Software can be used for executing the compression and decompression, the blending, and the duration and pitch control routines.

The foregoing description of preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Obviously, many modifications and variations within the scope of the claims will be apparent to practitioners skilled in this art. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims.

Claims

1. An apparatus for synthesizing speech in response to a sequence of sound segment codes representing speech, comprising memory (15) storing a set (25) of quantization vectors (QV_p) having shaped quantization noise spectra, said quantization vectors being generated by an inverse noise shaping filter operation performed on a first set (123) of quantization vectors (C_p) that correspond to the sound segment codes;

   means (200,10), responsive to sound segment codes in the sequence, for identifying (203) strings of quantization vectors in the set (125) of quantization vectors (QV_p) having shaped quantization noise spectra for respective sound segment codes in the sequence;
   means (10), coupled to the means for identifying and the memory (15), for generating (204,205,206) a speech data sequence in response to the strings of quantization vectors; and
   an audio transducer (16,17), coupled to the means for generating, to generate sound in response to the speech data sequence.

2. The apparatus of claim 1, wherein the sound segment codes comprise data encoded using the first set of quantization vectors, and the set (125) of quantization vectors (QV_p) having shaped quantization noise spectra is different from the first set (123) of quantization vectors (C_p) but related to it according to the noise shaping filter operation.

3. The apparatus of claim 1 or 2, wherein the first set of quantization vectors represent quantization of filtered sound segment data, and the means for generating a speech data sequence includes;
   means for applying an inverse filter to the identified strings of quantization vectors in generation of the speech data sequence.

4. The apparatus of claim 3, wherein the inverse filter includes parameters chosen so that any multiplies are replaced by shift and/or add operations in application of the inverse filter.

5. The apparatus of claim 1 or 2, wherein the first set of quantization vectors represent quantization of results of linear prediction filtering of sound segment data, and the means for generating a speech data sequence includes;
means for applying an inverse linear prediction filter to the identified strings of quantization vectors in generation of the speech data sequence.

6. The apparatus or claim 1 or 2 or 5, wherein the first set of quantization vectors represent quantization of results of pitch filtering of sound segment data, and the means for generating a speech data sequence includes;
means for applying an inverse pitch filter to the identified strings of quantization vectors in generation of the speech data sequence.

7. The apparatus of any preceding claim, wherein the means for generating a speech data sequence includes:
means for concatenating the identified strings of quantization vectors and supplying the concatenated strings for the speech data sequence.

8. The apparatus of any preceding claim, wherein the identified strings of quantization vectors each have a beginning and an ending, and means for generating a speech data sequence includes;

means for supplying the identified strings of quantization vectors for respective sound segment codes in sequence; and
means for blending the ending of an identified string of quantization vectors of a particular sound segment code in the sequence with the beginning of an identified string of quantization vectors of an adjacent sound segment code in the sequence to smooth discontinuities between the particular and adjacent sound segment codes in the speech data sequence.

9. The apparatus of any preceding claim, wherein the means for generating a speech data sequence includes;
means, responsive to the sound segment codes for adjusting pitch and duration of the identified strings of quantization vectors in the speech data sequence.

10. The apparatus of any preceding claim further including an encoder including:

a store for an encoding set of quantization vectors different from the set of quantization vectors used in decoding; and
means for generating the sound segment codes in response to the encoding set and sound segment data.

11. The apparatus of claim 10, wherein the encoder further includes a linear prediction filter.

12. The apparatus of claim 10 or 11, wherein the encoder further includes a pitch filter.

13. An apparatus for synthesizing speech in response to a text, comprising:

means for translating text to a sequence of sound segment codes;
means for generating a set (125) of quantization vectors (QV_p) having shaped quantization noise spectra by applying an inverse noise shaping filter function to a first set (123) of quantization vectors (C_p) that correspond to the sound segment codes;
memory (15) storing the set (125) of quantization vectors (QV_p) having shaped quantization noise spectra;
means (10), responsive to sound segment codes in the sequence, for identifying (203) strings of quantization vectors in the set (125) of quantization vectors (QV_p) having shaped quantization noise spectra for respective sound segment codes in the sequence;
means (10), coupled to the means for identifying and the memory (15), for generating (204,205,206) a speech data sequence in response to the strings of quantization vectors; and
an audio transducer (16, 17), coupled to the means for generating, to generate sound in response to the speech data sequence.

14. The apparatus of claim 13, wherein the sound segment codes comprise data encoded using a first set (123) of quantization vectors (C_p), and the set (125) of quantization vectors (QV_p) having shaped quantization noise spectra is different from the first set of quantization vectors (C_p) but related to it according to the noise shaping filter function.

15. The apparatus of claim 13 or 14, wherein the first set of quantization vectors represent quantization of filtered

sound segment data, and the means for generating a speech data sequence includes:

means for applying an inverse filter to the identified strings of quantization vectors in generation of the speech data sequence.

16. The apparatus of claim 15, wherein the inverse filter includes parameters chosen so that any multiplies are replaced by shift and/or add operations in application of the inverse filter.

17. The apparatus of claim 13, 14, 15 or 16, wherein the means for translating includes a table of encoded diphones, having entries including data identifying a string of quantization vectors in the set for respective diphones, and the sequence of sound segment codes comprises a sequence of indices to the table of encoded diphones representing the text; and

the means for identifying strings of quantization vectors includes means responsive to the sound segment codes for accessing the entries in the table of encoded diphones.

18. The apparatus of any of claims 13 to 17, wherein the first set of quantization vectors represent quantization of results of linear prediction filtering of sound segment data, and the means for generating a speech data sequence includes:

means for applying an inverse linear prediction filter to the identified strings of quantization vectors in generation of the speech data sequence.

19. The apparatus of any of claims 13 to 18, wherein the first set of quantization vectors represent quantization of results of pitch filtering of sound segment data, and the means for generating a speech data sequence includes:

means for applying an inverse pitch filter to the identified strings of quantization vectors in generation of the speech data sequence.

20. The apparatus of any of claims 13 to 19, wherein the means for generating a speech data sequence includes:

means for concatenating the identified strings of quantization vectors and supplying the concatenated strings for the speech data sequence.

21. The apparatus of any of claims 13 to 20, wherein the identified strings of quantization vectors each have a beginning and an ending, and means for generating a speech data sequence includes:

means for supplying the identified strings of quantization vectors for respective sound segment codes in sequence; and

means for blending the ending of an identified string of quantization vectors of a particular sound segment code in the sequence with the beginning of an identified string of quantization vectors of an adjacent sound segment code in the sequence to smooth discontinuities between the particular and adjacent sound segment codes in the speech data sequence.

22. The apparatus of any of claims 13 to 21, wherein the means for generating a speech data sequence includes:

means, responsive to the sound segment codes for adjusting pitch and duration of the identified strings of quantization vectors in the speech data sequence.

23. The apparatus of claim 21, further comprising:

means, responsive to the sound segment codes for adjusting pitch and duration of the identified strings of quantization vectors in the speech data sequence.

24. The apparatus of any of claims 13 to 23, further including an encoder including:

a store for an encoding set of quantization vectors different from the set of quantization vectors used in decoding; and

means for generating the sound segment codes in response to the encoding set and sound segment data.

25. The apparatus of claim 24, wherein the encoder further includes a linear prediction filter.

26. The apparatus of claim 24 or 25, wherein the encoder further includes a pitch filter.

27. An apparatus for synthesizing speech in response to a text, comprising :

a programmable processor (10) to execute routines to produce a speech data sequence;
an audio transducer (16,17) coupled to the processor, to generate sound in response to the speech data sequence;
a table memory (15) coupled to the processor, storing a noise-shaped set (125) of quantization vectors (QV_p) produced by performing an inverse noise shaping filter operation on a first set (123) of quantization vectors, and a table of encoded diphones having entries including data identifying (23) a string of quantization vectors (QV_p) in the said noise-shaped set (125) for respective diphones; and
an instruction memory (15), coupled to the processor, storing a translator routine for execution by the processor to translate (21) the text to a sequence of diphone indices, and a decoder routine for execution by the processor including

means, responsive to diphone indices in the sequence, for accessing the table of encoded diphones to identify strings of quantization vectors (QV_p) in the said noise-shaped set (125) for diphones in the text; and
means, coupled to the means for accessing and the table memory, for retrieving the identified strings of quantization vectors (QV_p);
means, coupled with the means for retrieving, for producing diphone data strings in response to the identified strings of quantization vectors, wherein the diphone data strings each have a beginning and an ending;
means, coupled to the means for producing, for blending (24) the ending of a particular diphone data string in the sequence with the beginning of an adjacent diphone data string in the sequence to smooth discontinuities between the particular and adjacent diphone data strings to produce a smoothed string of quantized speech data; and
means, responsive to the text and the smoothed string of quantized speech data; for adjusting (25, 26) pitch and duration of the identified strings of quantization vectors for the diphones in the sequence to produce the speech data sequence for supply to the audio transducer.

28. The apparatus of claim 27, wherein the data identifying a string of quantization vectors comprise data encoded using the first set (123) of quantization vectors (C_p) and the set (125) of noise compensated quantization vectors (QV_p) is different from the first set (123) of quantization vectors (C_p) but related to it according to the noise shaping filter operation.

29. The apparatus of claim 27, wherein the first set of quantization vectors represent quantization of filtered sound segment data, and the means for producing diphone data strings includes:
means for applying an inverse filter to the identified strings of quantization vectors.

30. The apparatus of claim 29, wherein the inverse filter includes parameters chosen so that any multiplies are replaced by shift and/or add operations in application of the inverse filter.

31. The apparatus of claim 27, 28, 29 or 30, wherein the first set of quantization vectors represent quantization of results of linear prediction filtering of sound segment data, and the means for producing diphone data strings includes:
means for applying a inverse linear prediction filter to the identified strings of quantization vectors.

32. The apparatus of any of claims 27 to 31, wherein the first set of quantization vectors represent quantization of results of pitch filtering of sound segment data, and the means for producing diphone data strings includes:
means for applying an inverse pitch filter to the identified strings of quantization vectors.

Patentansprüche

1. Vorrichtung zum Synthetisieren von Sprache als Reaktion auf eine Folge von Sprache darstellenden Klangsegmentcodes, die aufweist: einen Speicher (15), der einen Satz (125) von Quantisierungsvektoren (QVp) speichert, die geformte Quantisierungsrauschspektren aufweisen, wobei die Quantisierungsvektoren durch eine inverse Rauschformungsfilteroperation erzeugt werden, die an einem ersten Satz (123) von Quantisierungsvektoren (Cp) ausgeführt wird, die den Klangsegmentcodes entsprechen;

Einrichtungen (299,10) die auf Klangsegmentcodes in der Folge ansprechen, zum Identifizieren (203) von Ketten von Quantisierungsvektoren im Satz (125) von Quantisierungsvektoren (QVp), die geformte Quanti-

sierungsrauschspektren für jeweilige Klangsegmentcodes in der Folge aufweisen;
Einrichtungen (10), die an die Einrichtungen zum Identifizieren und den Speicher (15) gekoppelt sind, zum Erzeugen (204,205,206) einer Sprachdatenfolge als Reaktion auf die Ketten von Quantisierungsvektoren; und
einen Audiotransducer (16,17), der an die Einrichtungen zum Erzeugen gekoppelt ist, um Klang als Reaktion auf die Sprachdatenfolge zu erzeugen.

2. Vorrichtung nach Anspruch 1, wobei die Klangsegmentcodes Daten aufweisen, die unter Verwendung des ersten Satzes von Quantisierungsvektoren codiert werden, und der Satz (125) von Quantisierungsvektoren (QVp), die geformte Quantisierungsrauschspektren aufweisen, vom ersten Satz (123) von Quantisierungsvektoren (Cp) verschieden ist, jedoch mit ihm entsprechend der Rauschformungsfilteroperation in Beziehung steht.

3. Vorrichtung nach Anspruch 1 oder 2, wobei der erste Satz von Quantisierungsvektoren eine Quantisierung von gefilterten Klangsegmentdaten darstellt, und die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:
Einrichtungen zum Anwenden eines inversen Filters auf die identifizierten Ketten von Quantisierungsvektoren bei Erzeugung der Sprachdatenfolge.

4. Vorrichtung nach Anspruch 3, wobei der inverse Filter Parameter aufweist, die so gewählt werden, daß alle Multiplikationen bei Anwendung des inversen Filters durch Schiebe-und/oder Addieroperationen ersetzt werden.

5. Vorrichtung nach Anspruch 1 oder 2, wobei der erste Satz von Quantisierungsvektoren eine Quantisierung von Ergebnissen einer linearen Vorhersagefilterung von Klangsegmentdaten darstellt und die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen;
Einrichtungen zum Anwenden eines inversen linearen Vorher' sagefilters auf die identifizierten Ketten von Quantisierungsvektoren bei Erzeugung der Sprachdatenfolge.

6. Vorrichtung nach Anspruch 1 oder 2 oder 5, wobei der erste Satz von Quantisierungsvektoren eine Quantisierung von Ergebnissen einer Tonlagenfilterung von Klangsegmentdaten darstellt, und die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:
Einrichtungen zum Anwenden eines inversen Tonlagenfilters auf die identifizierten Ketten von Quantisierungsvektoren bei Erzeugung der Sprachdatenfolge.

7. Vorrichtung nach einem der vorhergehenden Ansprüche, wobei die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:
Einrichtungen zum Verketten der identifizierten Ketten von Quantisierungsvektoren und Liefern der verketteten Ketten für die Sprachdatenfolge.

8. Vorrichtung nach einem der vorhergehenden Ansprüche, wobei die identifizierten Ketten von Quantisierungsvektoren jeweils einen Anfang und ein Ende aufweisen, und Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:

Einrichtungen zum Liefern der identifizierten Ketten von Quantisierungsvektoren für jeweilige Klangsegmentcodes in einer Folge; und
Einrichtungen zum Mischen des Endes einer identifizierten Kette von Quantisierungsvektoren eines bestimmten Klangsegmentcodes in der Folge mit dem Anfang einer identifizierten Kette von Quantisierungsvektoren eines angrenzenden Klangsegmentcodes in der Folge, um Diskontinuitäten zwischen dem bestimmten und angrenzenden Klangsegmentcode in der Sprachdatenfolge zu glätten.

9. Vorrichtung nach einem der vorhergehenden Ansprüche, wobei die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:
Einrichtungen, die auf die Klangsegmentcodes ansprechen, zum Einstellen von Tonlage und Dauer der identifizierten Ketten von Quantisierungsvektoren in der Sprachdatenfolge.

10. Vorrichtung nach einem der vorhergehenden Ansprüche, die ferner einen Codierer aufweist, der aufweist:

einen Speicher für einen Codiersatz von Quantisierungsvektoren, der vom Satz von Quantisierungsvektoren verschieden ist, der beim Decodieren verwendet wird; und
Einrichtungen zum Erzeugen der Klangsegmentcodes als Reaktion auf den Codiersatz und Klangsegmentdaten.

11. Vorrichtung nach Anspruch 10, wobei der Codierer ferner einen lineareren Vorhersagefilter aufweist.

12. Vorrichtung nach Anspruch 10 oder 11, wobei der Codierer ferner einen Tonlagenfilter aufweist.

13. Vorrichtung zum Synthetisieren von Sprache als Reaktion auf einen Text, die aufweist:

Einrichtungen zum Übersetzen von Text in eine Folge von Klangsegmentcodes;
Einrichtungen zum Erzeugen eines Satzes (125) von Quantisierungsvektoren (QVp), die geformte Quantisierungsrauschspektren aufweisen, durch Anwenden einer inversen Rauschformungsfilterfunktion auf einen ersten Satz (123) von Quantisierungsvektoren (Cp), die den Klangsegmentcodes entsprechen;
einen Speicher (15), der den Satz (125) von Quantisierungsvektoren (QVp) speichert, die geformte Quantisierungsrauschspektren aufweisen;
Einrichtungen (10), die auf Klangsegmentcodes in der Folge ansprechen, zum Identifizieren (203) von Ketten von Quantisierungsvektoren in dem Satz (125) von Quantisierungsvektoren (QVp), die geformte Quantisierungsrauschspektren aufweisen, für jeweilige Klangsegmentcodes in der Folge;
Einrichtungen (10), die an die Einrichtungen zum Identifizieren und den Speicher (15) gekoppelt sind, zum Erzeugen (204,205,206) einer Sprachdatenfolge als Reaktion auf die Ketten von Quantisierungsvektoren; und einen Audiotransducer (16,17), der an die Einrichtungen zum Erzeugen gekoppelt ist, um Klang als Reaktion auf die Sprachdatenfolge zu erzeugen.

14. Vorrichtung nach Anspruch 13, wobei die Klangsegmentcodes Daten aufweisen, die unter Verwendung eines ersten Satzes (123) von Quantisierungsvektoren (Cp) codiert werden, und der Satz (125) von Quantisierungsvektoren (Qvp), die geformte Quantisierungsrauschspektren aufweisen, vom ersten Satz von Quantisierungsvektoren (Cp) verschieden ist, jedoch mit ihm entsprechend der Rauschformungsfilteroperation in Beziehung steht.

15. Vorrichtung nach Anspruch 13 oder 14, wobei der erste Satz von Quantisierungsvektoren eine Quantisierung von gefilterten Klangsegmentdaten darstellt, und die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen: Einrichtungen zum Anwenden eines inversen Filters auf die identifizierten Ketten von Quantisierungsvektoren bei Erzeugung der Sprachdatenfolge.

16. Vorrichtung nach Anspruch 15, wobei der inverse Filter Parameter aufweist, die so gewählt werden, daß alle Multiplikationen bei Anwendung des inversen Filters durch Schiebe-und/oder Addieroperationen ersetzt werden.

17. Vorrichtung nach Anspruch 13, 14, 15 oder 16, wobei die Einrichtungen zum Übersetzen eine Tabelle von codierten Diphonen aufweisen, die Einträge aufweist, die Daten aufweisen, die eine Kette von Quantisierungsvektoren in dem Satz für jeweilige Diphone aufweisen, und die Folge von Klangsegmentcodes eine Folge von Indizes auf die Tabelle von codierten Diphonen, die den Text darstellen, aufweist; und die Einrichtungen zum Identifizieren von Ketten von Quantisierungsvektoren Einrichtungen aufweisen, die auf die Klangsegmentcodes ansprechen, zum Zugreifen auf die Einträge in der Tabelle von codierten Diphonen.

18. Vorrichtung nach einem der Ansprüche 13 bis 17, wobei der erste Satz von Quantisierungsvektoren eine Quantisierung von Ergebnissen einer linearen Vorhersagefilterung von Klangsegmentdaten darstellt, und die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:
Einrichtungen zum Anwenden eines inversen linearen Vorhersagefilters auf die identifizierten Ketten von Quantisierungsvektoren bei Erzeugung der Sprachdatenfolge.

19. Vorrichtung nach einem der Ansprüche 13 bis 18, wobei der erste Satz von Quantisierungvektoren eine Quantisierung von Ergebnissen einer Tonlagenfilterung von Klangsegmentdaten darstellt, und die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:
Einrichtungen zum Anwenden eines inversen Tonlagenfilters auf die identifizierten Ketten von Quantisierungsvektoren bei Erzeugung der Sprachdatenfolge.

20. Vorrichtung nach einem der Ansprüche 13 bis 19, wobei die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:
Einrichtungen zum Verketten der identifizierten Ketten von Quantisierungsvektoren und Liefern der verketteten Ketten für die Sprachdatenfolge.

21. Vorrichtung nach einem der Ansprüche 13 bis 20, wobei die identifizierten Ketten von Quantisierungsvektoren

jeweils einen Anfang und ein Ende aufweisen, und Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:

Einrichtungen zum Liefern der identifizierten Ketten von Quantisierungsvektoren für jeweilige Klangsegmentcodes in einer Folge; und
Einrichtungen zum Mischen des Endes einer identifizierten Kette von Quantisierungsvektoren eines bestimmten Klangsegmentcodes in der Folge mit dem Anfang einer identifizierten Kette von Quantisierungsvektoren eines angrenzenden Klangsegmentcodes in der Folge, um Diskontinuitäten zwischen dem bestimmten und angrenzenden Klangsegmentcode in der Sprachdatenfolge zu glätten.

22. Vorrichtung nach einem der Ansprüche 13 bis 21, wobei die Einrichtungen zum Erzeugen einer Sprachdatenfolge aufweisen:
Einrichtungen, die ansprechen auf die Klangsegmentcodes, zum Einstellen einer Tonlage und Dauer der identifizierten Ketten von Quantisierungsvektoren in der Sprachdatenfolge.

23. Vorrichtung nach Anspruch 21, die ferner aufweist:
Einrichtungen, die ansprechen auf die Klangsegmentcodes, zum Einstellen einer Tonlage und Dauer der identifizierten Ketten von Quantisierungsvektoren in der Sprachdatenfolge.

24. Vorrichtung nach einem der Ansprüche 13 bis 23, die ferner einen Codierer aufweist, der aufweist:

einen Speicher für einen Codiersatz von Quantisierungsvektoren, der vom Satz von Quantisierungsvektoren verschieden ist, der beim Decodieren verwendet wird; und
Einrichtungen zum Erzeugen der Klangsegmentcodes als Reaktion auf den Codiersatz und Klangsegmentdaten.

25. Vorrichtung nach Anspruch 24, wobei der Codierer ferner einen linearen Vorhersagefilter aufweist.

26. Vorrichtung nach Anspruch 24 oder 25, wobei der Codierer ferner einen Tonlagenfilter aufweist.

27. Vorrichtung zum Synthetisieren von Sprache als Reaktion auf einen Text, die aufweist:

einen programmierbaren Prozessor (10), um Routinen auszuführen, um eine Sprachdatenfolge zu erzeugen;
einen Audiotransducer (16,17), der an den Prozessor gekoppelt ist, um Klang als Reaktion auf die Sprachdatenfolge zu erzeugen;
einen Tabellenspeicher (15), der an den Prozessor gekoppelt ist, der einen rauschgeformten Satz (125) von Quantisierungsvektoren (QVp), die durch Ausführen einer inversen Rauschformungsfilteroperation am ersten Satz (123) von Quantisierungsvektoren erzeugt werden, und eine Tabelle von codierten Diphonen speichert, die Einträge aufweist, die Daten aufweisen, die eine Kette von Quantisierungsvektoren (QVp) in dem rauschgeformten Satz (125) für jeweilige Diphone identifizieren (23);
und einen Befehlsspeicher (15), der an den Prozessor gekoppelt ist, der eine Übersetzerroutine zur Ausführung durch den Prozessor, um den Text in eine Folge von Diphonindizes zu übersetzen (21), und eine Decodierroutine zur Ausführung durch den Prozessor speichert, die aufweist

Einrichtungen, die auf Diphonindizes in der Folge ansprechen, zum Zugreifen auf die Tabelle von codierten Diphonen, um Ketten von Quantisierungsvektoren (QVp) in dem rauschgeformten Satz (125) für Diphone im Text zu identifizieren; und
Einrichtungen, die an die Einrichtungen zum Zugreifen und den Tabellenspeicher gekoppelt sind, zum Wiedergewinnen der identifizierten Ketten von Quantisierungsvektoren (QVp);
Einrichtungen, die mit den Einrichtungen zum Wiedergewinnen gekoppelt sind, zum Erzeugen von Diphondatenketten als Reaktion auf die identifizierten Ketten von Quantisierungsvektoren, wobei die Diphondatenketten jeweils einen Anfang und ein Ende aufweisen;
Einrichtungen, die an die Einrichtungen zum Erzeugen gekoppelt sind, zum Mischen (24) des Endes einer bestimmten Diphondatenkette in der Folge mit dem Anfang einer angrenzenden Diphondatenkette in der Folge, um Diskontinuitäten zwischen der bestimmten und angrenzenden Diphondatenkette zu glätten, um eine geglättete Kette von quantisierten Sprachdaten zu erzeugen; und
Einrichtungen, die auf den Text und die geglättete Kette von quantisierten Sprachdaten ansprechen, zum Einstellen (25,26) einer Tonlage und Dauer der identifizierten Ketten von Quantisierungsvektoren für die

Diphone in der Folge, um die Sprachdatenfolge zur Lieferung an den Audio wandler zu erzeugen.

28. Vorrichtung nach Anspruch 27, wobei die Daten, die eine Kette von Quantisierungsvektoren identifizieren, Daten aufweisen, die unter Verwendung des ersten Satzes (123) von Quantisierungsvektoren (Cp) codiert werden, und der Satz (125) von rauschkompensierten Quantisierungsvektoren (QVp) vom ersten Satz (123) von Quantisierungsvektoren (Cp) verschieden ist, jedoch mit ihm entsprechend der Rauschformungsfilteroperation in Beziehung steht.

29. Vorrichtung nach Anspruch 27, wobei der erste Satz von Quantisierungsvektoren eine Quantisierung von gefilterten Klangsegmentdaten darstellt, und die Einrichtungen zum Erzeugen von Diphondatenketten aufweisen:
Einrichtungen zum Anwenden eines inversen Filters auf die identifizierten Ketten von Quantisierungsvektoren.

30. Vorrichtung nach Anspruch 29, wobei der inverse Filter Parameter aufweist, die so gewählt werden, daß alle Multiplikationen bei Anwendung des inversen Filters durch Schiebe-und/oder Addieroperationen ersetzt werden.

31. Vorrichtung nach Anspruch 27, 28, 29 oder 30, wobei der erste Satz von Quantisierungsvektoren eine Quantisierung von Ergebnissen einer linearen Vorhersagefilterung von Klangsegmentdaten darstellt, und die Einrichtungen zum Erzeugen von Diphondatenketten aufweisen:
Einrichtungen zum Anwenden eines inversen linearen Vorhersagefilters auf die identifizierten Ketten von Quantisierungsvektoren.

32. Vorrichtung nach einem der Ansprüche 27 bis 31, wobei der erste Satz von Quantisierungsvektoren eine Quantisierung von Ergebnissen einer Tonlagenfilterung von Klangsegmentdaten darstellt, und die Einrichtungen zum Erzeugen von Diphondatenketten aufweisen:
Einrichtungen zum Anwenden eines inversen Tonlagenfilters auf die identifizierten Ketten von Quantisierungsvektoren.

Revendications

1. Dispositif pour synthétiser de la parole en réponse à une séquence de codes de segments de son représentant de la parole, comprenant une mémoire (15) enregistrant un ensemble (125) de vecteurs de quantification (QV_p) ayant des spectres de bruit de quantification mis en forme, les vecteurs de quantification étant produits par une opération de filtrage de mise en forme de bruit inverse réalisée sur un premier ensemble (123) de vecteurs de quantification (C_p) qui correspondent aux codes de segments de son ;

   des moyens (299, 10), sensibles à des codes de segments de son dans la séquence, pour identifier (203) des chaînes de vecteurs de quantification dans l'ensemble (125) de vecteurs de quantification (QV_p) ayant des spectres de bruit de quantification mis en forme pour des codes de segments de son respectifs dans la séquence ;
   des moyens (10), couplés aux moyens d'identification et à la mémoire (15), pour produire (204, 205, 206) une séquence de données de parole en réponse aux chaînes de vecteurs de quantification ; et
   un transducteur (16, 17) audio, couplé aux moyens de production, pour produire un son en réponse à la séquence de données de parole.

2. Dispositif suivant la revendication 1, dans lequel les codes de segments de son comprennent des données codées en utilisant le premier ensemble de vecteurs de quantification, et l'ensemble (125) de vecteurs de quantification (QV_p) ayant des spectres de bruit de quantification mis en forme est différent du premier ensemble (123) de vecteurs de quantification (C_p) mais est lié à celui-ci conformément à l'opération de filtrage de mise en forme du bruit.

3. Dispositif suivant la revendication 1 ou 2, dans lequel le premier ensemble de vecteurs de quantification représente la quantification de données de segments de son filtrées, et les moyens de production d'une séquence de données de parole comportent :
   des moyens pour appliquer un filtre inverse aux chaînes identifiées de vecteurs de quantification lors de la production de la séquence de données de parole.

4. Dispositif suivant la revendication 3, dans lequel le filtre inverse comporte des paramètres choisis de façon que d'éventuelles multiplications soient remplacées par des opérations de décalage et/ ou d'addition dans l'application

du filtre inverse.

5. Dispositif suivant la revendication 1 ou 2, dans lequel le premier ensemble de vecteurs de quantification représente la quantification de résultats d'un filtrage de prédiction linéaire de données de segments de son, et les moyens de production d'une séquence de données de parole comportent :

des moyens pour appliquer un filtrc de prédiction linéaire inverse aux chaînes identifiées de vecteurs de quantification dans la production de la séquence de données de parole.

6. Dispositif suivant la revendication : 1 ou 2 ou 5, dans lequel le premier ensemble de vecteurs de quantification représente la quantification de résultais du filtrage de la hauteur de son de données de ségments de son, et les moyens de production d'une séquence de données de parole comportent ;

des moyens pour appliquer un filtre de hauteur de son inverse aux chaînes identifiées de vecteurs de quantification dans la production de la séquence de données de parole.

7. Dispositif suivant l'une quelconque des revendications précédentes, dans lequel les moyens destinés à produire une séquence de données de parole comportent :

des moyens pour concaténer les chaînes identifiées de vecteurs de quantification et fournir les chaînes concaténées pour la séquence de données de parole.

8. Dispositif suivant l'une quelconque des revendications précédentes, dans lequel les chaînes identifiées de vecteurs de quantification ont chacune un début et une fin, et les moyens destinés à produire une séquence de données de parole comportent :

des moyens pour fournir les chaînes identifiées de vecteurs de quantification pour des codes de segments de son respectifs selon une séquence ; ete

des moyens pour mélanger la fin d'une chaîne identifiée de vecteurs de quantification d'un code de segments de son particuliers dans la séquence avec le début d'une chaîne identifiée de vecteurs de quantification d'un code de segment de son adjacent, dans la séquence, pour lisser les discontinuités entre les codes de segments de son particulier et adjacent dans la séquence de données de parole.

9. Dispositif suivant l'une quelconque des revendications précédentes, dans lequel les moyens destinés à produire une séquence de données de parole comportent :

des moyens sensibles aux codes de segments de son pour ajuster la hauteur de son et la durée des chaînes identifiées de vecteurs de quantification dans la séquence de données de parole.

10. Dispositif suivant l'une quelconque des revendications précédentes, comportant en outre un codeur comportant :

une mémoire destinée à un ensemble de codage de vecteurs de quantification différent de l'ensemble de vecteurs de quantification utilisés dans le décodage ; et

des moyens pour produire les codes de segments de son en réponse à l'ensemble de codage et aux de segments de son.

11. Dispositif suivant la revendication 10, dans lequel le codeur comporte en outre un filtre de prédiction linéaire.

12. Dispositif suivant la revendication 10 ou 11, dans lequel le codeur comporte en outre un filtre de hauteur de son.

13. Dispositif pour synthétiser de la parole en réponse à un texte, comprenant :

des moyens pour traduire du texte en une séquence de codes de segments de son ;

des moyens destinés à produire un ensemble (125) de vecteurs de quantification (QV_p) ayant des spectres de bruit de quantification mis en forme par application d'une fonction de filtrage de mise en forme de bruit inverse à un premier ensemble (123) de vecteurs de quantification (C_p) qui correspondent aux codes de segments de son ;

une mémoire (15) mémorisant l'ensemble (125) de vecteurs de quantification (QV_p) ayant des spectres de bruit de quantification mis en forme ;

des moyens (10), sensibles à des codes de segments de son dans la séquence, pour identifier (203) des chaînes de vecteurs de quantification dans l'ensemble (125) de vecteurs de quantification (QV_p) ayant des spectres de bruit de quantification mis en formz pour des codes de segments de son respectifs dans la sé-

quence;
des moyens (10), couplés aux moyens d'identification et à la mémoire (15), pour produire (204, 205, 206) une séquence de données de parole en réponse aux chaînes de vecteurs de quantification ; et
un transducteur (16, 17) audio, couplé aux moyens de production, pour produire du son en réponse à une séquence de données de parole.

14. Dispositif suivant la revendication 13, dans lequel les codes de segments de son comprennent des données codées en utilisant un premier ensemble (123) de vecteurs de quantification (C_p), et l'ensemble (125) de vecteurs de quantification (QV_p) ayant des spectres de bruit de quantification mis en forme est différent du premier ensemble de vecteurs de quantification (C_p) mais est lié à celui-ci conformément à la fonction de filtrage de mise en forme du bruit.

15. Dispositif suivant la revendication 13 ou 14, dans lequel le premier ensemble de vecteurs de quantification représente la quantification de données de segments de son filtrées et les moyens destinés à produire une séquence de données de parole comportent :
des moyens pour appliquer un filtre inverse aux chaînes identifiées de vecteurs de quantification lors de la production de la séquence de données de parole.

16. Dispositif suivant la revendication 15, dans lequel le filtre inverse comporte des paramètres choisis de façon que d'éventuelles multiplications soient remplacées par des opérations de décalage et/ou d'addition dans l'application du filtre inverse.

17. Dispositif suivant la revendication 13, 14, 15 ou 16, dans lequel les moyens de traduction comportent un tableau de diphones codés, ayant des entrées comportant des données identifiant une chaîne de vecteurs de quantification dans l'ensemble pour des diphones respectifs, et la séquence de codes de segments de son comprend une séquence d'indices pointant sur le tableau de diphones codés représentant le texte ; et
les moyens d'identification de chaînes de vecteurs de quantification comportent des moyens sensibles aux codes de segments de son pour accéder aux entrées dans le tableau de diphones codés.

18. Dispositif suivant l'une quelconque des revendications 13 à 17, dans lequel le premier ensemble de vecteurs de quantification représente la quantification de résultats du filtrage de prédiction linéaire de données de segments de son, et les moyens destinés à produire une séquence de données de parole comportent :
des moyens pour appliquer un filtre de prédiction linéaire inverse aux chaînes identifiées de vecteurs de quantification lors de la production de la séquence de données de parole.

19. Dispositif suivant l'une quelconque des revendications 13 à 18, dans lequel le premier ensemble de vecteurs de quantification représente la quantification de résultats du filtrage de la hauteur de son de données de segments de son, et les moyens destinés à produire une séquence de données de parole comportent :
des moyens destinés à appliquer un filtre de hauteur de son inverse aux chaînes identifiées de vecteurs de quantification lors de la production de la séquence de données de parole.

20. Dispositif suivant l' une quelconque des revendications 13 à 19, dans lequel les moyens destinés à produire une séquence de données de parole comportent :
des moyens pour concaténer les chaînes identifiées de vecteurs de quantification et fournir les chaînes concaténées pour la séquence de données de parole.

21. Dispositif suivant l'une quelconque des revendications 13 à 20, dans lequel les chaînes identifiées de vecteurs de quantification ont chacune un début et une fin, et les moyens destinés à produire une séquence de données de parole comportent :

des moyens pour fournir les chaînes identifiées de vecteurs de quantification pour des codes de segments de son respectifs en une séquence ; et
des moyens pour mélanger la fin d'une chaîne identifiée de vecteurs de quantification d'un code de segment de son particulier dans la séquence avec le début d'une chaîne identifiée de vecteurs de quantification d'un code de segment de son adjacent dans la séquence pour lisser des discontinuités entre les codes de segments de son particulier et adjacent dans la séquence de données de parole.

22. Dispositif suivant l'une quelconque des revendications 13 à 21, dans lequel les moyens destinés à produire une

séquence de données de parole comportent :

des moyens, sensibles aux codes de segments de son pour ajuster la hauteur de son et la durée des chaînes identifiées de vecteurs de quantification dans la séquence de données de parole.

23. Dispositif suivant la revendication 21, comprenant en outre :

des moyens sensibles aux codes de segments de son pour ajuster la hauteur de son et la durée des chaînes identifiées de vecteurs de quantification dans la séquence de données de parole.

24. Dispositif suivant l'une quelconque des revendications 13 à 23, comportant en outre un codeur comportant :

une mémoire destinée à un ensemble de codage de vecteurs de quantification différent de l'ensemble de vecteurs de quantification utilisé dans le décodage ; et
des moyens pour générer les codes de segments de son en réponse à l'ensemble de codage et aux données de segments de son.

25. Dispositif suivant la revendication 24, dans lequel le codeur comporte en outre un filtre de prédiction linéaire.

26. Dispositif suivant la revendicaition 24 ou 25, dans lequel le codeur comporte en outre un filtre de hauteur de son.

27. Dispositif destiné à synthétiser de la parole en réponse à un texte, comprenant :

un processeur (10) programmable pour exécuter des sous-programmes pour produire une séquence de données de parole ;
un transducteur (16, l'Il audio, couplé au processeur, pour produire du son en réponse à la séquence de données de parole;
une mémoire (15) de tableau, couplée au processeur, enregistrant un ensemble (125) à bruit mis en forme de vecteurs de quantification (QV_p) produits en réalisant une opération de filtrage de mise en forme de bruit inverse sur un premier ensemble (123) de vecteurs de quantification, et un tableau de diphones codés avant des entrées comportant des données identifiant (23) une chaîne de vecteurs de quantification (QV_p) dans l'ensemble (125) à bruit mis en forme destiné aux diphones respectifs ; et
une mémoire (15) d'instructions couplée au processeur, enregistrant un sous-programme traducteur destiné à être exécuté par le processeur pour traduire (21) le texte en une séquence d'indices de diphones, et un sous-programme décodeur destiné à être exécuté par le processeur, comportant :

des moyens, sensibles à des indices de diphones dans la séquence, pour accéder au tableau de diphones codés afin d'identifier des chaînes de vecteurs de quantification (QV_p) dans l'ensemble (125) à bruit mis en forme destiné à des diphones dans le texte ; et
des moyens, couplés aux moyens d'accès et de mémoire de tableau, pour extraire les chaînes identifiées de vecteurs de quantification (QV_p) ;
des moyens, couplés aux moyens d'extraction, pour produire des chaînes de données de diphones en réponse aux chaînes identifiées de vecteurs de quantification, dans lequel les chaînes de données de diphones ont chacune un début et une fin ;
des moyens, couplés aux moyens de production, pour mélanger (24) la fin d'une chaîne de données de diphones particulières dans la séquence avec le début d'une chaîne de données de diphones adjacente dans la séquence pour lisser les discontinuités entre les chaînes de données de diphones particulière et adjacente afin de produire une chaîne lissée de données de parole quantifiées ; et
des moyens, sensibles au texte et à la chaîne lissée de données de parole quantifiées, pour ajuster (25, 26) la hauteur de son et la durée des chaînes identifiées de vecteurs de quantification pour les diphones de la séquence afin de produire la séquence de données de parole destinée à être fournie au transducteur audio.

28. Dispositif suivant la revendication 27, dans lequel les données identifiant une chaîne de vecteurs de quantification comprennent des données codées en utilisant le premier ensemble (123) de vecteurs de quantification (C_p), et l'ensemble (125) de vecteurs de quantification (QV_p) à bruit compensé est différent du premier ensemble (123) de vecteurs de quantification (C_p), mais est lié a celui-ci conformément à l'opération de filtrage de mise en forme du bruit.

29. Dispositif suivant la revendication 27, dans lequel le premier ensemble de vecteurs de quantification représente

la quantification de données de segments de son filtrés, et les moyens destinés à produire des chaînes de données de diphones comportent :

des moyens pour appliquer un filtre inverse aux chaînes identifiées de vecteurs de quantification.

30. Dispositif suivant la revendication 29, dans lequel le filtre inverse comporte des paramètres choisis de facon que d'éventuelles multiplications soient remplacées par des opérations de décalage et/ou d'addition lors de l'application du filtre inverse.

31. Dispositif suivant la revendication 27, 28, 29 ou 30, dans lequel le premier ensemble de vecteurs de quantification représente la quantification de résultats d'un filtrage de prédiction linéaire de données de segments de son, et les moyens destinés à produire des chaînes de données de diphones comportent :

des moyens destinés à appliquer un filtre de prédiction linéaire inverse aux chaînes identifiées de vecteurs de quantification.

32. Dispositif suivant l'une quelconque des revendications 27 à 31, dans lequel le premier ensemble de vecteurs de quantification représente la quantification de résultats du filtrage de hauteur de son de données de segments de son, et les moyens destinés à produire des chaînes de données de diphones comportent :

des moyens destinés à appliquer un filtre de hauteur de son inverse aux chaînes identifiées de vecteurs de quantification.

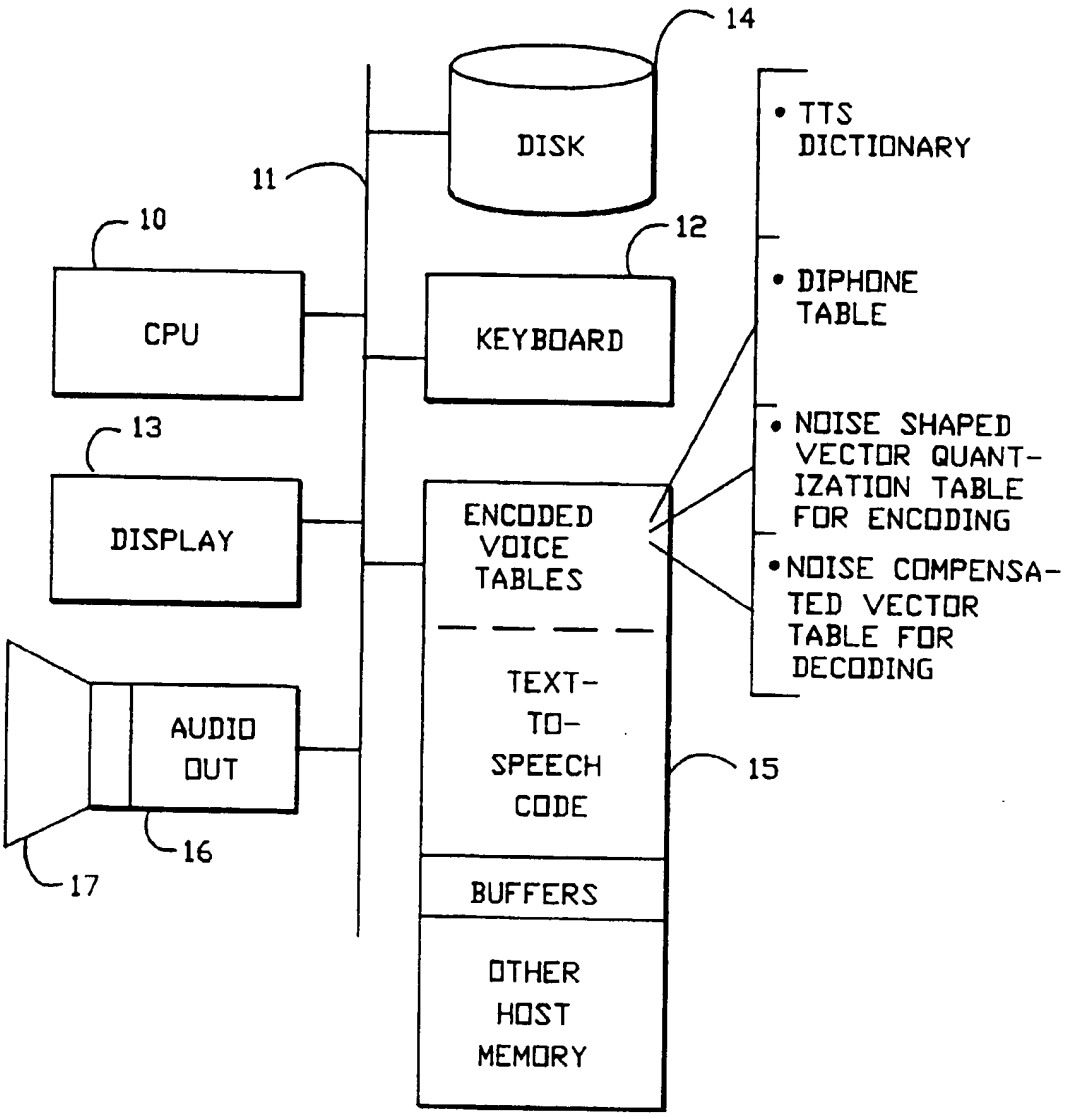
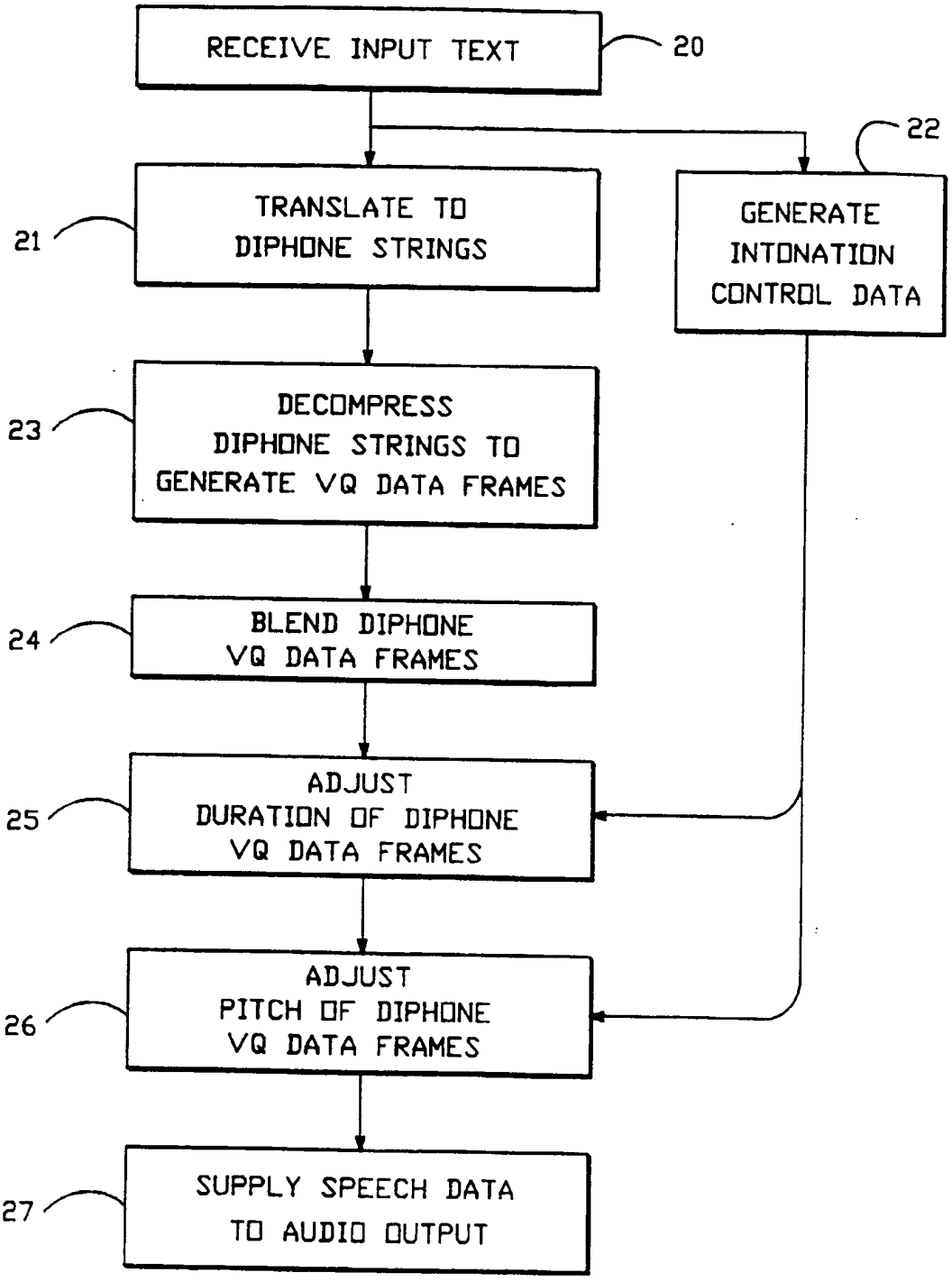

14
DISK
• TTS DICTIONARY
11
10
12
CPU
KEYBOARD
• DIPHONE TABLE
13
• NOISE SHAPED VECTOR QUANT-IZATION TABLE FOR ENCODING
DISPLAY
ENCODED VOICE TABLES
• NOISE COMPENSA-TED VECTOR TABLE FOR DECODING
TEXT-TO-SPEECH CODE
AUDIO OUT
15
16
17
BUFFERS
OTHER HOST MEMORY

FIG.—1



TEXT - TO - SPEECH CODE

FIG.-2

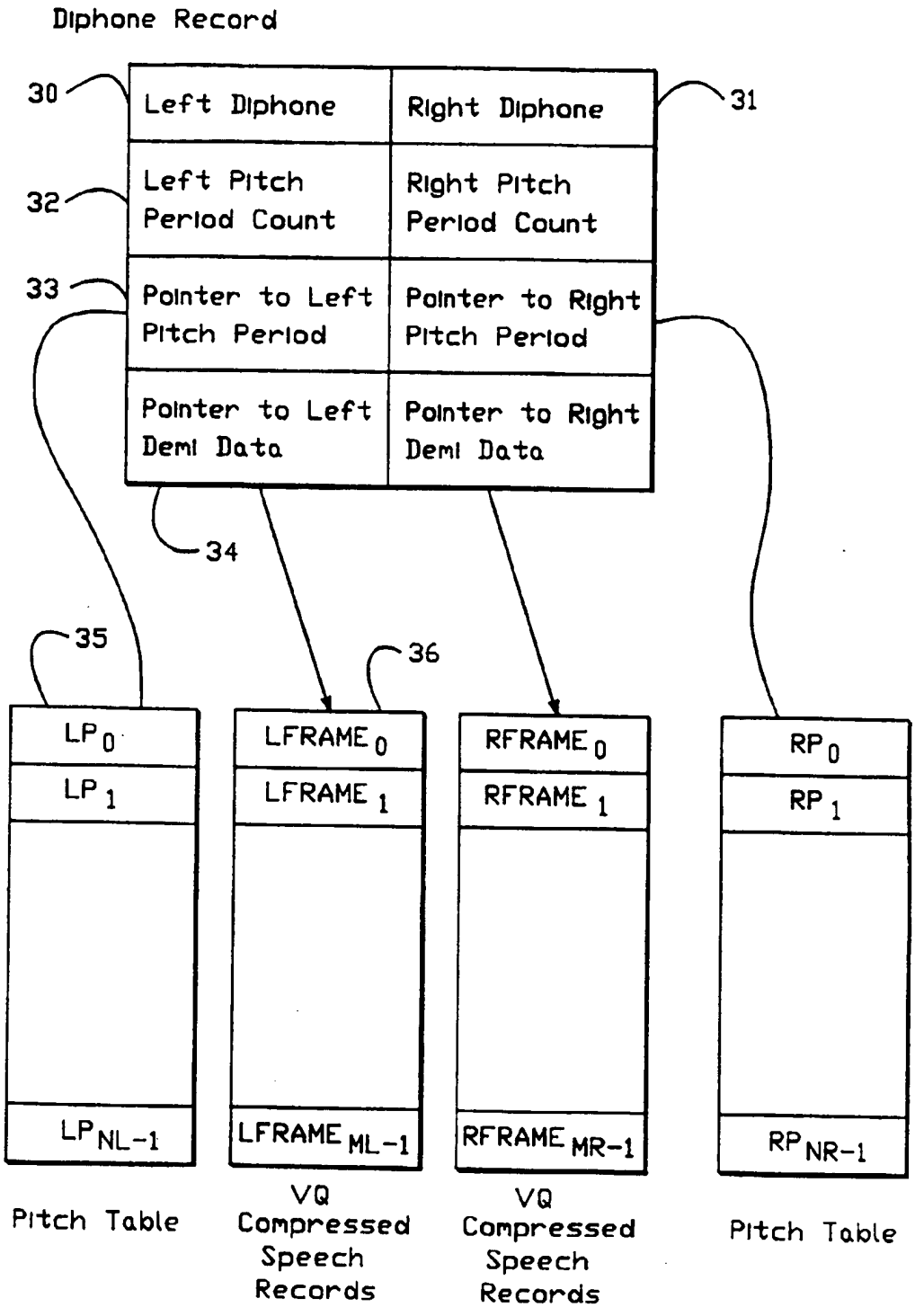

Diphone Record
30
31
32
33
34
35
36
Left Diphone
Right Diphone
Left Pitch Period Count
Right Pitch Period Count
Pointer to Left Pitch Period
Pointer to Right Pitch Period
Pointer to Left Demi Data
Pointer to Right Demi Data
LP 0
LP 1
LP NL-1
LFRAME 0
LFRAME 1
LFRAME ML-1
RFRAME 0
RFRAME 1
RFRAME MR-1
RP 0
RP 1
RP NR-1
Pitch Table
VQ Compressed Speech Records
VQ Compressed Speech Records
Pitch Table

FIG.-3

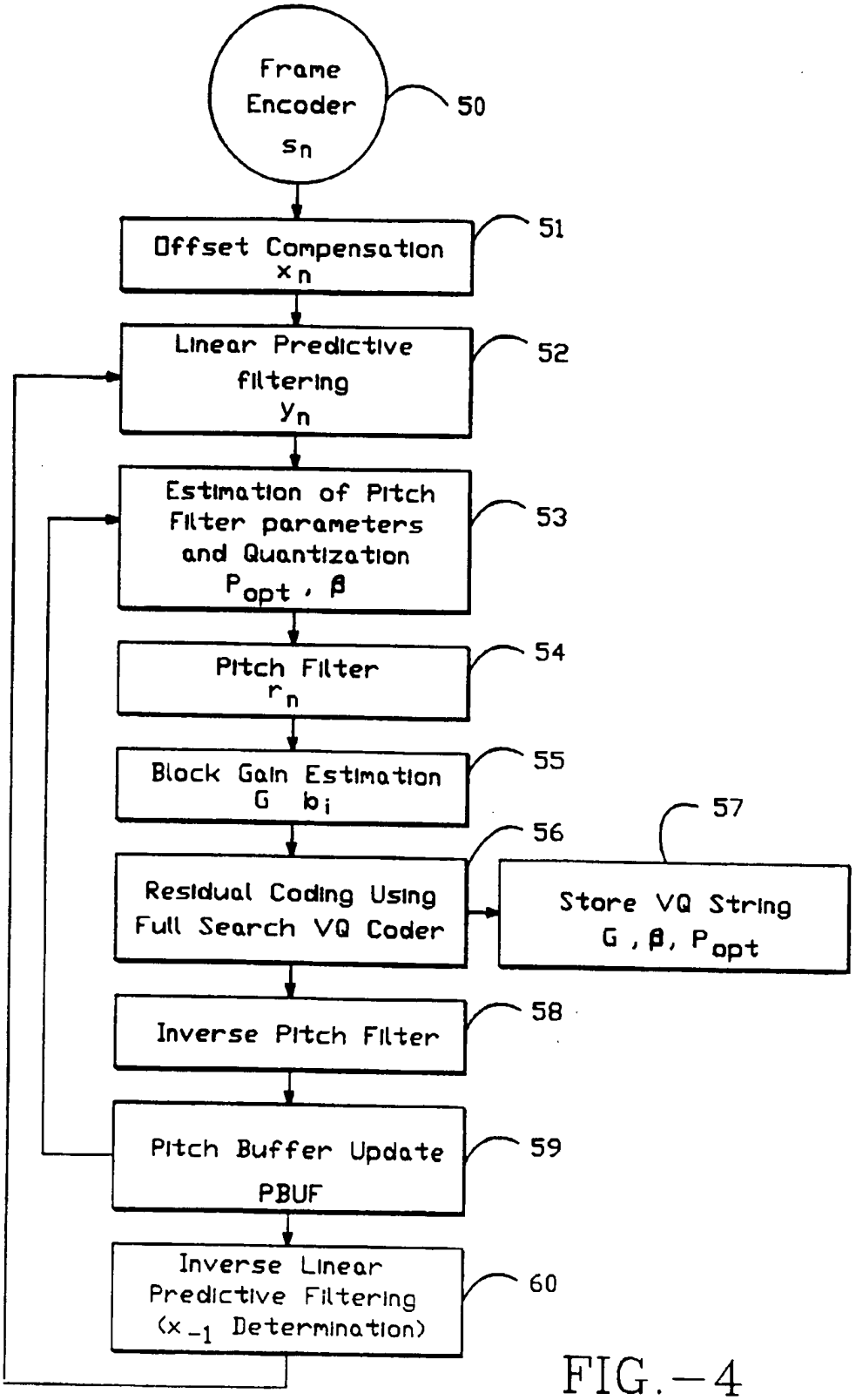

Frame Encoder s_n
50
Offset Compensation x_n
51
Linear Predictive filtering y_n
52
Estimation of Pitch Filter parameters and Quantization P_{opt} , β
53
Pitch Filter r_n
54
Block Gain Estimation G b_i
55
Residual Coding Using Full Search VQ Coder
56
Store VQ String G , β, P_{opt}
57
Inverse Pitch Filter
58
Pitch Buffer Update PBUF
59
Inverse Linear Predictive Filtering (x_{-1} Determination)
60

FIG.—4

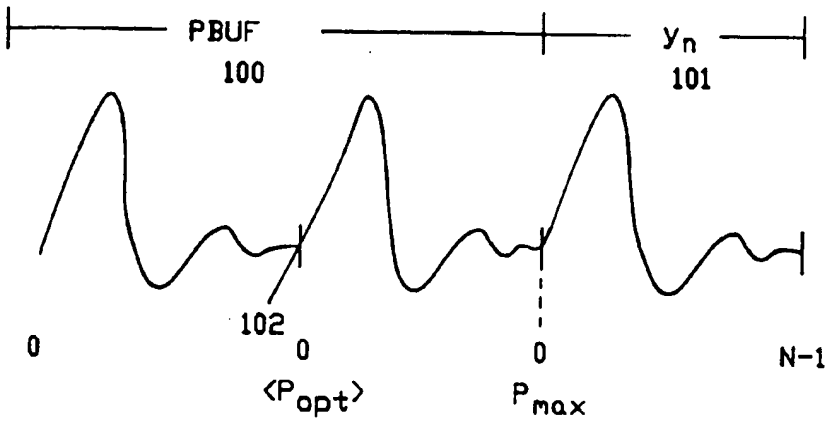

PBUF
100
y_n
101
102
0
0
0
N-1
<P_{opt}>
P_{max}

FIG.-5

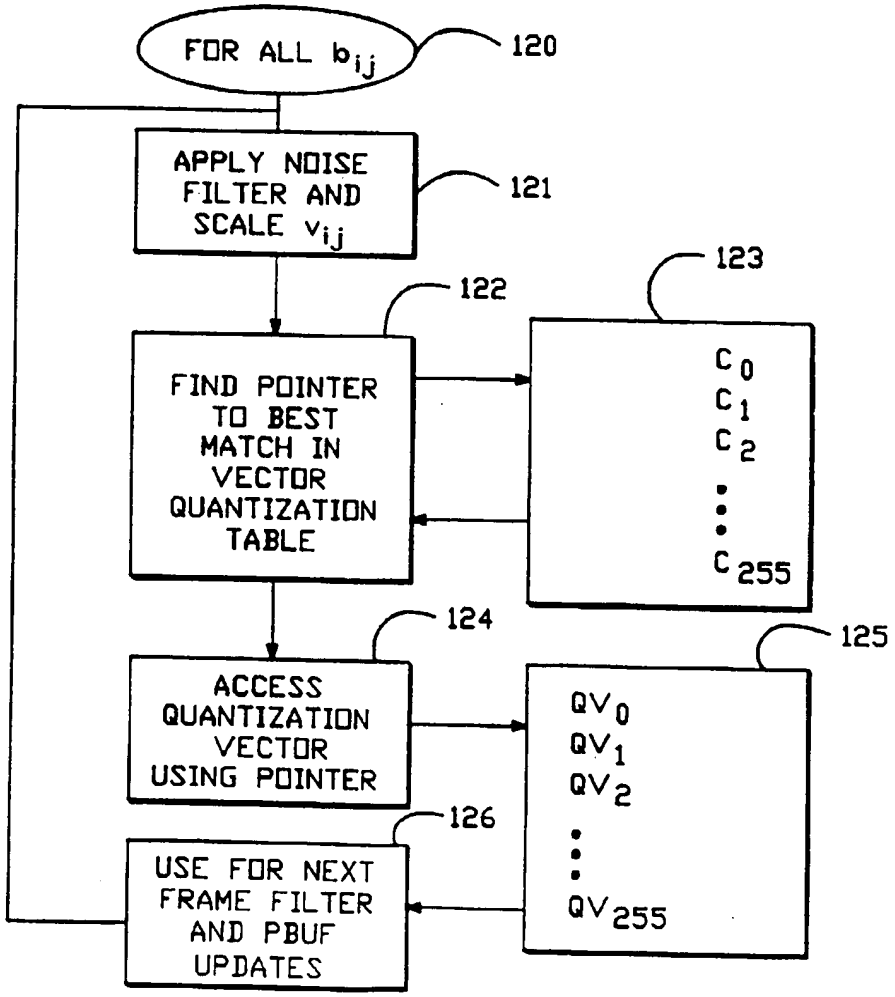

FOR ALL b_{ij}
120
APPLY NOISE FILTER AND SCALE v_{ij}
121
122
123
FIND POINTER TO BEST MATCH IN VECTOR QUANTIZATION TABLE
C_0
C_1
C_2
C_{255}
124
125
ACCESS QUANTIZATION VECTOR USING POINTER
QV_0
QV_1
QV_2
QV_{255}
126
USE FOR NEXT FRAME FILTER AND PBUF UPDATES

FIG.-6

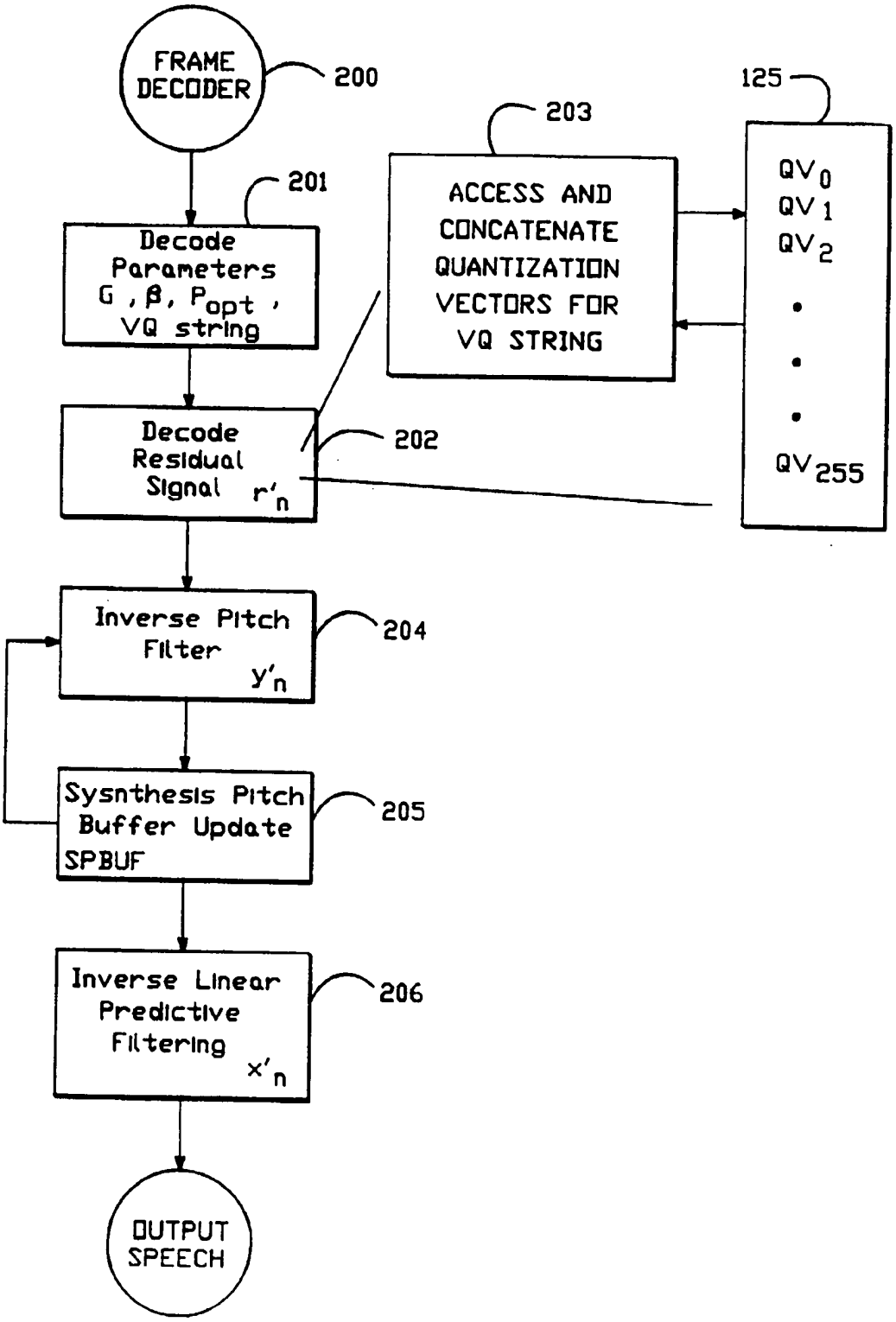

FRAME DECODER
200
Decode Parameters G , β, P_{opt} , VQ string
201
ACCESS AND CONCATENATE QUANTIZATION VECTORS FOR VQ STRING
203
125
QV_0
QV_1
QV_2
QV_{255}
Decode Residual Signal r'_n
202
Inverse Pitch Filter y'_n
204
Sysnthesis Pitch Buffer Update SPBUF
205
Inverse Linear Predictive Filtering x'_n
206
OUTPUT SPEECH

FIG.—7

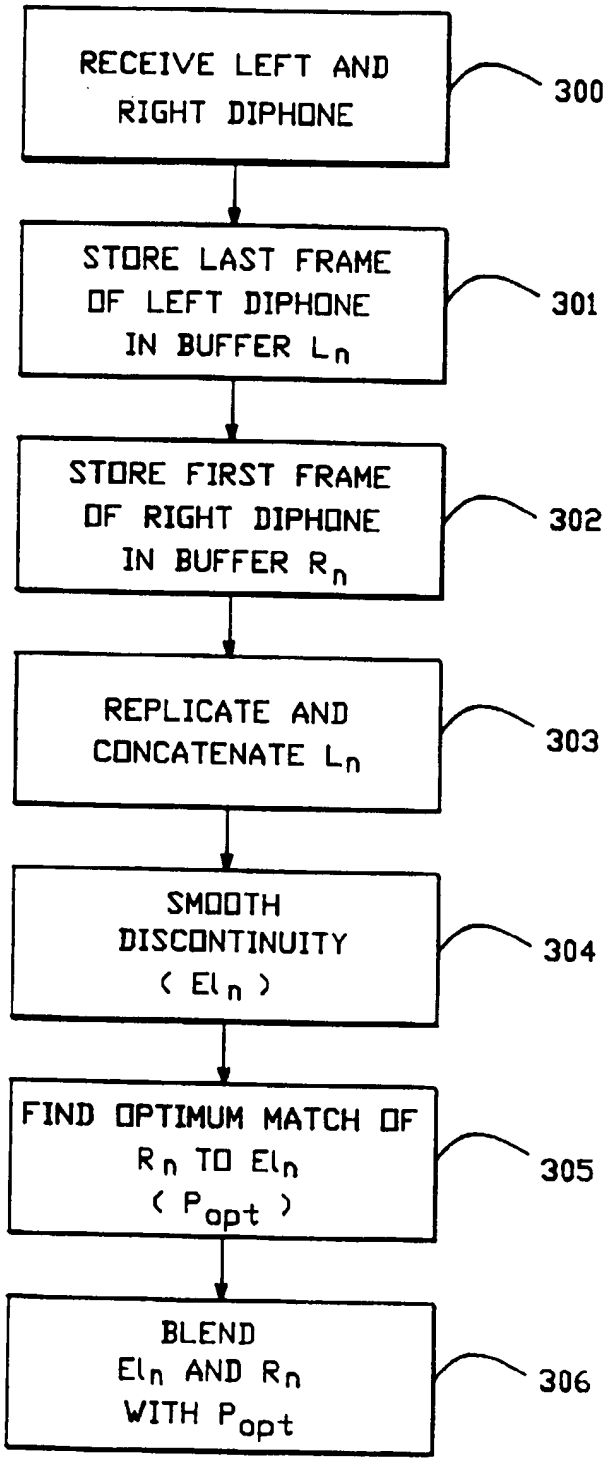

RECEIVE LEFT AND RIGHT DIPHONE
300
STORE LAST FRAME OF LEFT DIPHONE IN BUFFER L_n
301
STORE FIRST FRAME OF RIGHT DIPHONE IN BUFFER R_n
302
REPLICATE AND CONCATENATE L_n
303
SMOOTH DISCONTINUITY (El_n)
304
FIND OPTIMUM MATCH OF R_n TO El_n (P_{opt})
305
BLEND El_n AND R_n WITH P_{opt}
306

FIG.—8

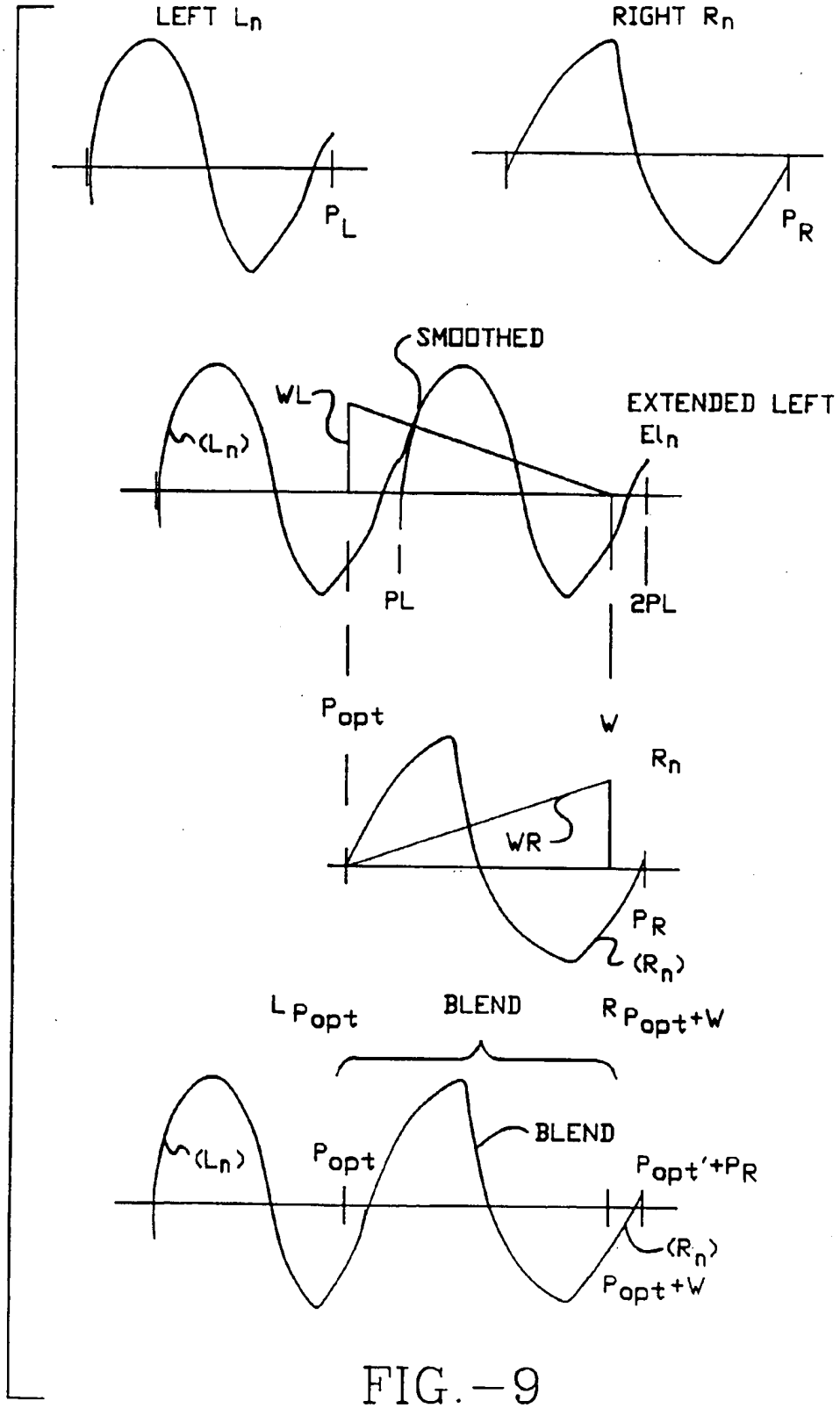
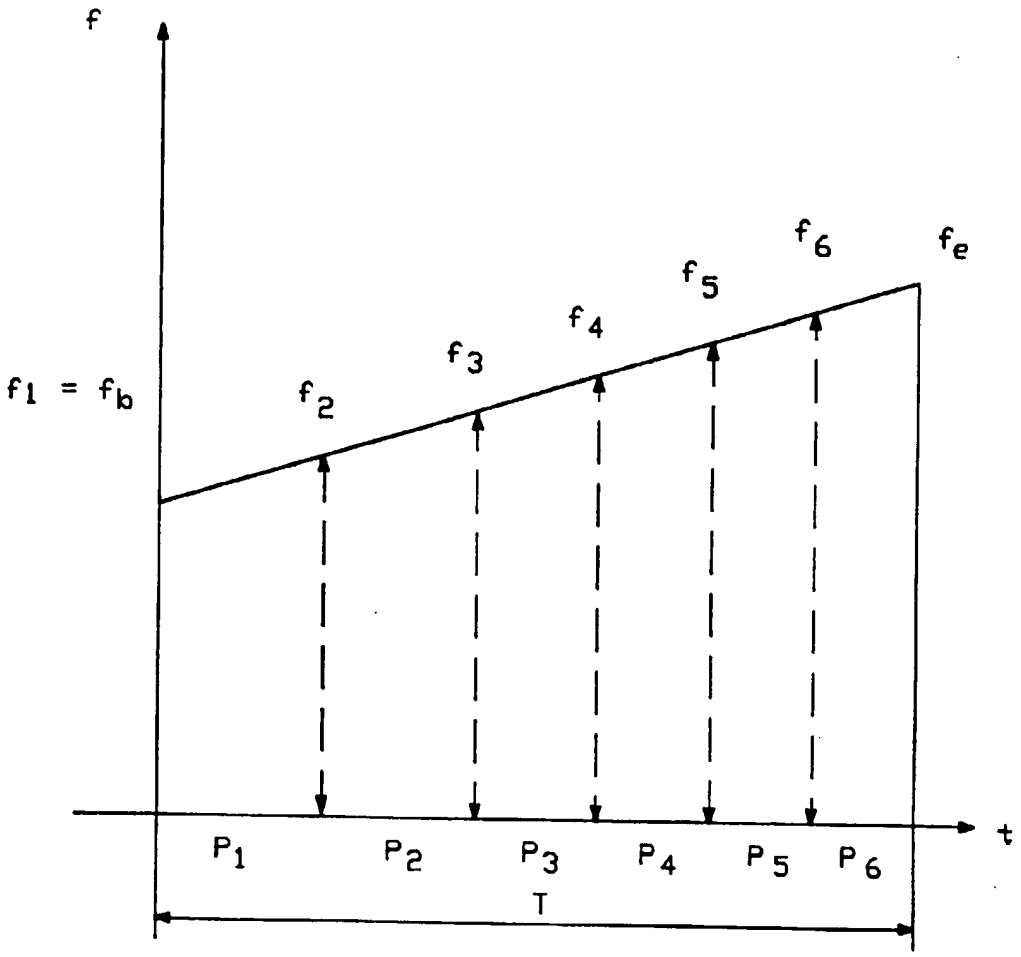

LEFT L_n
RIGHT R_n
P_L
P_R
SMOOTHED
WL
EXTENDED LEFT
El_n
(L_n)
PL
2PL
P_{opt}
W
R_n
WR
P_R
(R_n)
$L P_{opt}$
BLEND
$R P_{opt+W}$
BLEND
(L_n)
P_{opt}
$P_{opt'}+P_R$
(R_n)
P_{opt+W}

FIG.—9



NOTES:

T = Desired duration of a phoneme

f_b = Desired Begining Pitch in Hz

f_e = Desired Ending Pitch in Hz

P1, P2, ..., P6 are the desired pitch period in No. of Samples corresponding to the frequencies f1,f2,...f6.

Relationship between Pi and fi:

Pi = Fs/fi, where Fs is the Sampling frequency.

FIG.—10

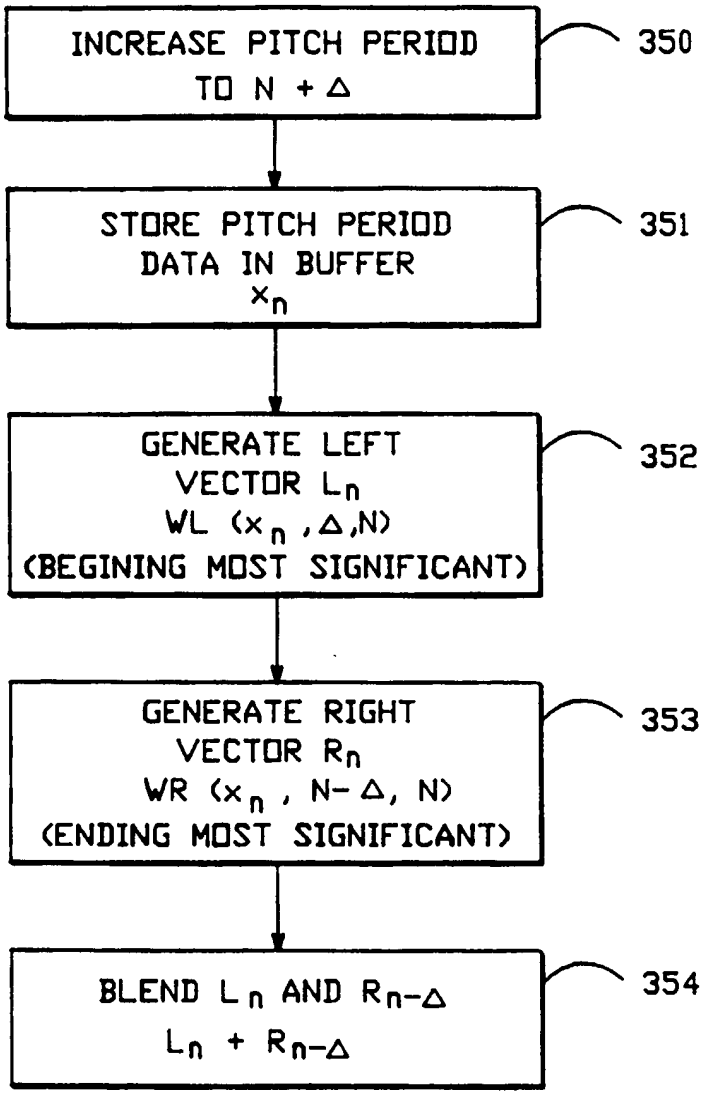

INCREASE PITCH PERIOD TO N + Δ
350
STORE PITCH PERIOD DATA IN BUFFER x_n
351
GENERATE LEFT VECTOR L_n WL (x_n, Δ, N) (BEGINING MOST SIGNIFICANT)
352
GENERATE RIGHT VECTOR R_n WR (x_n, N−Δ, N) (ENDING MOST SIGNIFICANT)
353
BLEND L_n AND $R_{n-\Delta}$ $L_n + R_{n-\Delta}$
354

FIG.—11

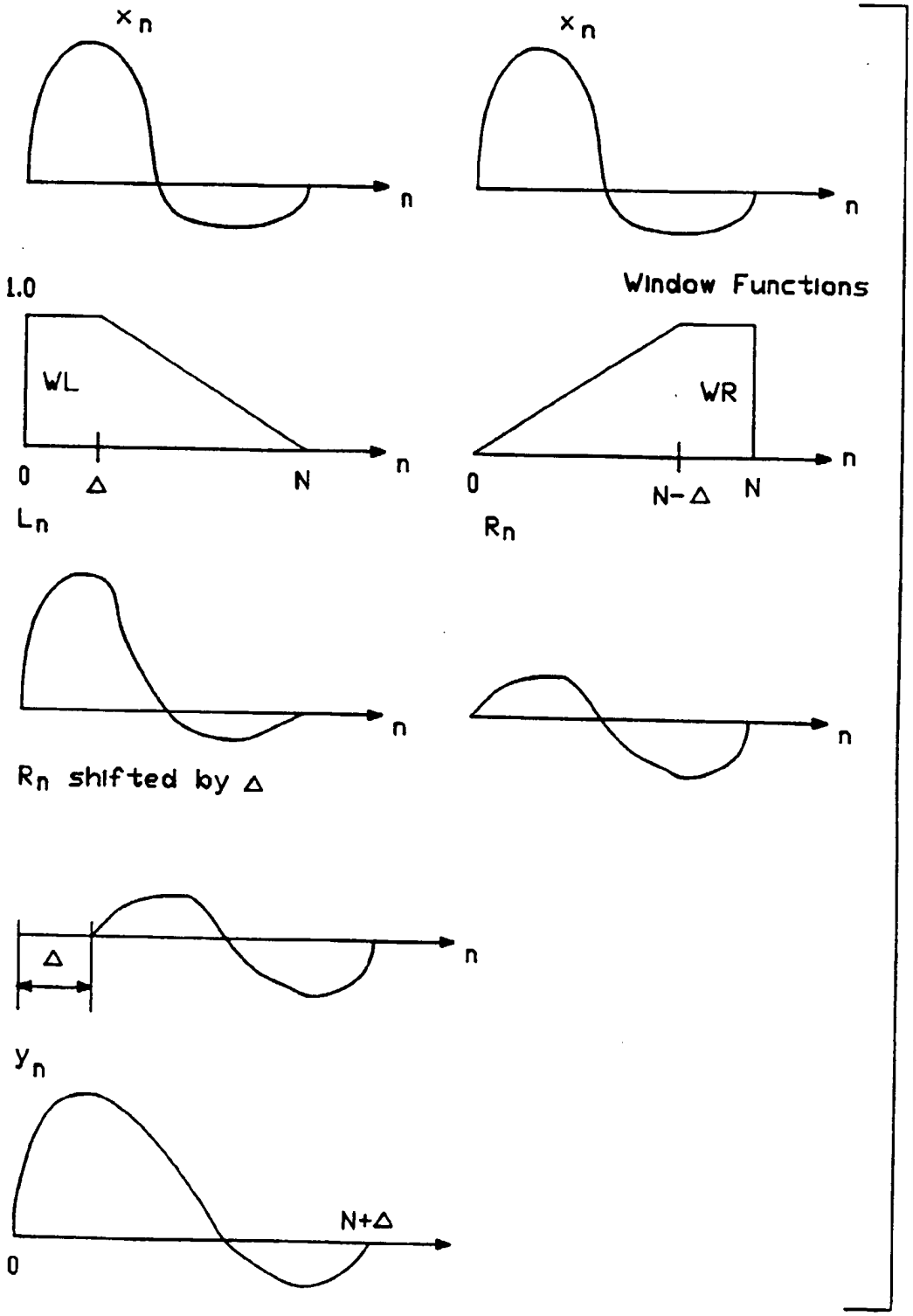

x n
n
x n
n
1.0
Window Functions
WL
0
Δ
N
n
Ln
WR
0
N-Δ
N
n
Rn
n
n
Rn shifted by Δ
Δ
n
yn
N+Δ
0

FIG.—12

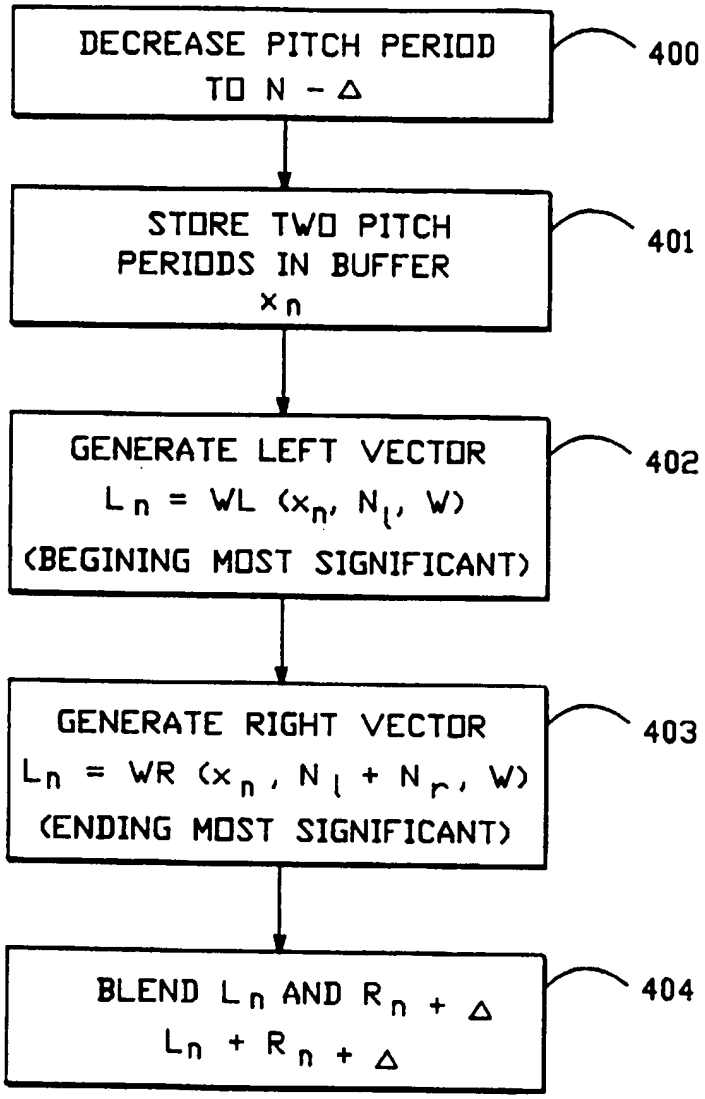

DECREASE PITCH PERIOD TO N - Δ
400
STORE TWO PITCH PERIODS IN BUFFER x_n
401
GENERATE LEFT VECTOR $L_n = WL(x_n, N_l, W)$ (BEGINING MOST SIGNIFICANT)
402
GENERATE RIGHT VECTOR $L_n = WR(x_n, N_l + N_r, W)$ (ENDING MOST SIGNIFICANT)
403
BLEND L_n AND $R_{n+\Delta}$ $L_n + R_{n+\Delta}$
404

FIG.—13

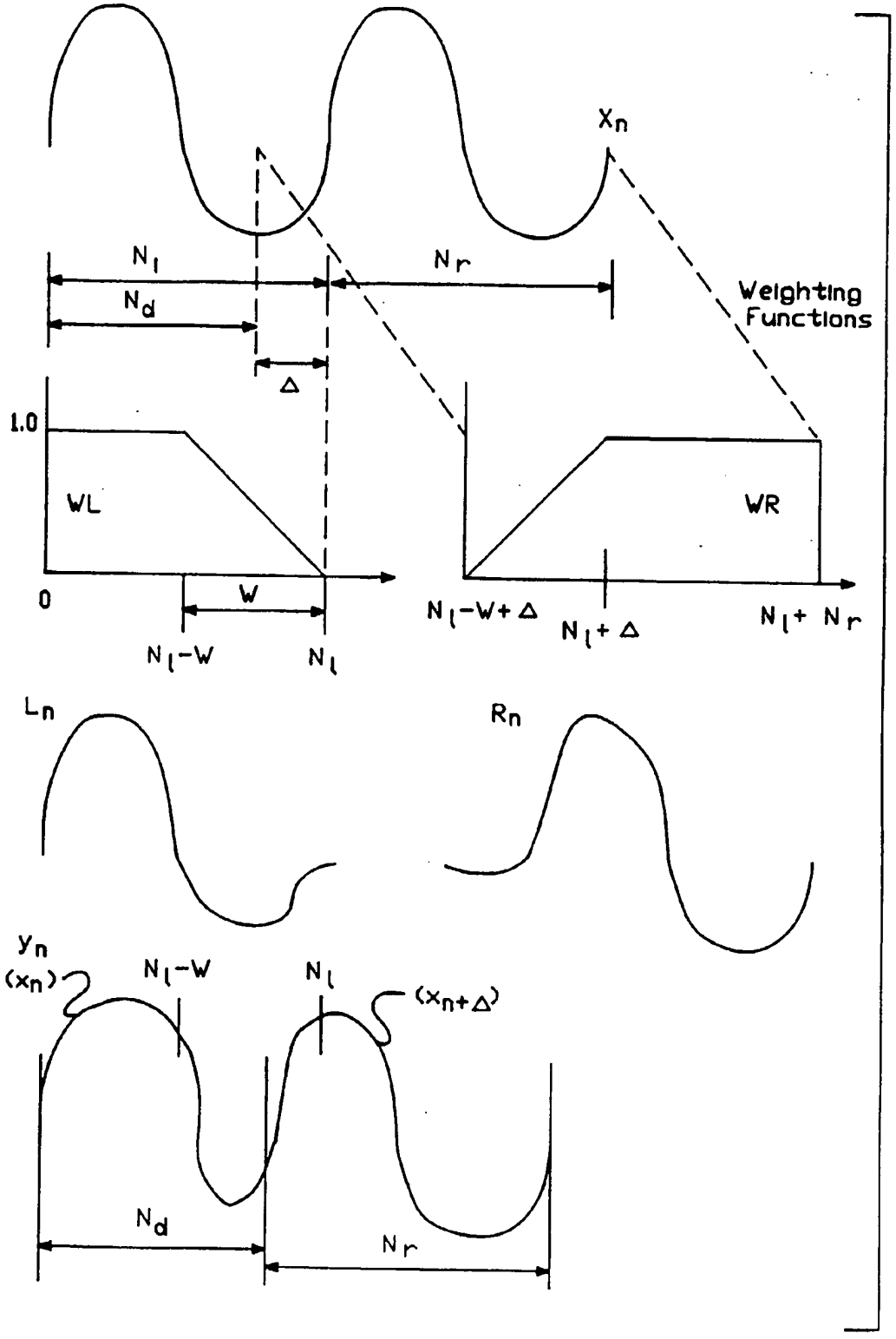

Xn
Nl
Nr
Nd
Δ
Weighting Functions
1.0
WL
WR
0
W
Nl-W
Nl
Nl-W+Δ
Nl+Δ
Nl+Nr
Ln
Rn
yn
(xn)
Nl-W
Nl
(xn+Δ)
Nd
Nr

FIG.—14

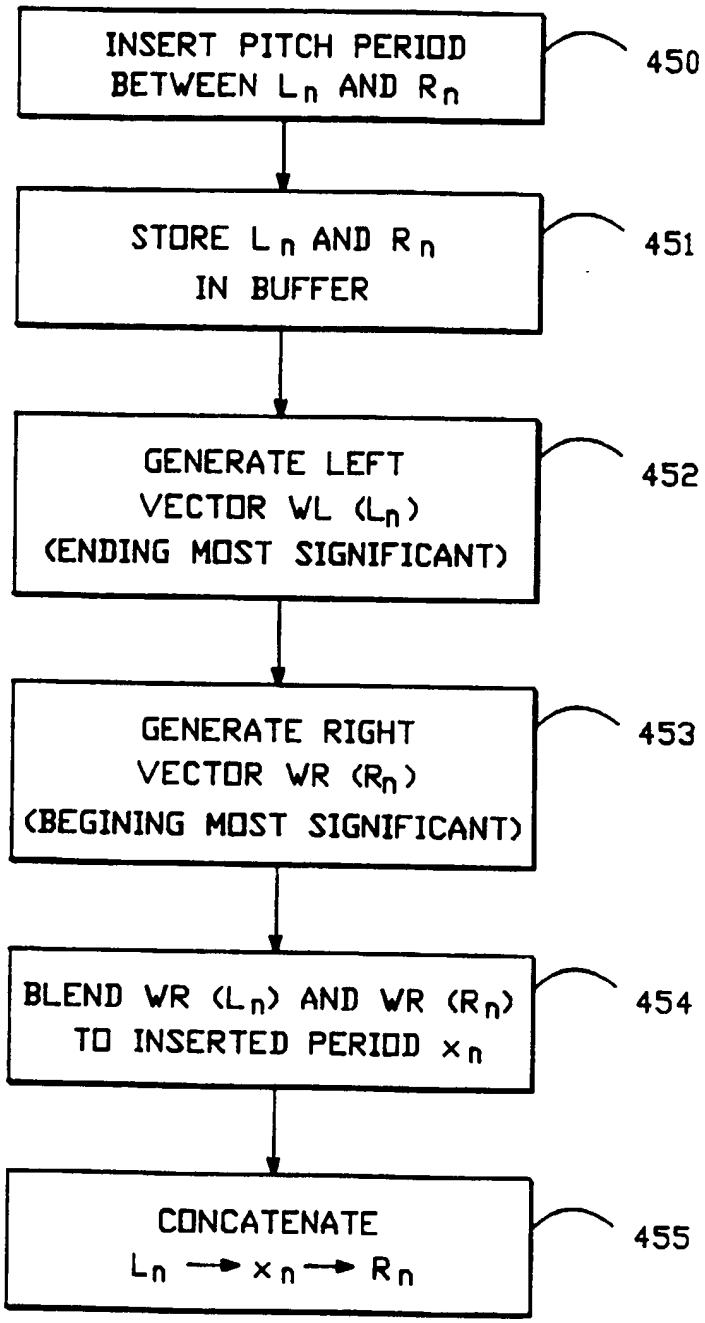

INSERT PITCH PERIOD BETWEEN L_n AND R_n
450
STORE L_n AND R_n IN BUFFER
451
GENERATE LEFT VECTOR WL (L_n) (ENDING MOST SIGNIFICANT)
452
GENERATE RIGHT VECTOR WR (R_n) (BEGINING MOST SIGNIFICANT)
453
BLEND WR (L_n) AND WR (R_n) TO INSERTED PERIOD x_n
454
CONCATENATE $L_n \rightarrow x_n \rightarrow R_n$
455

FIG.—15

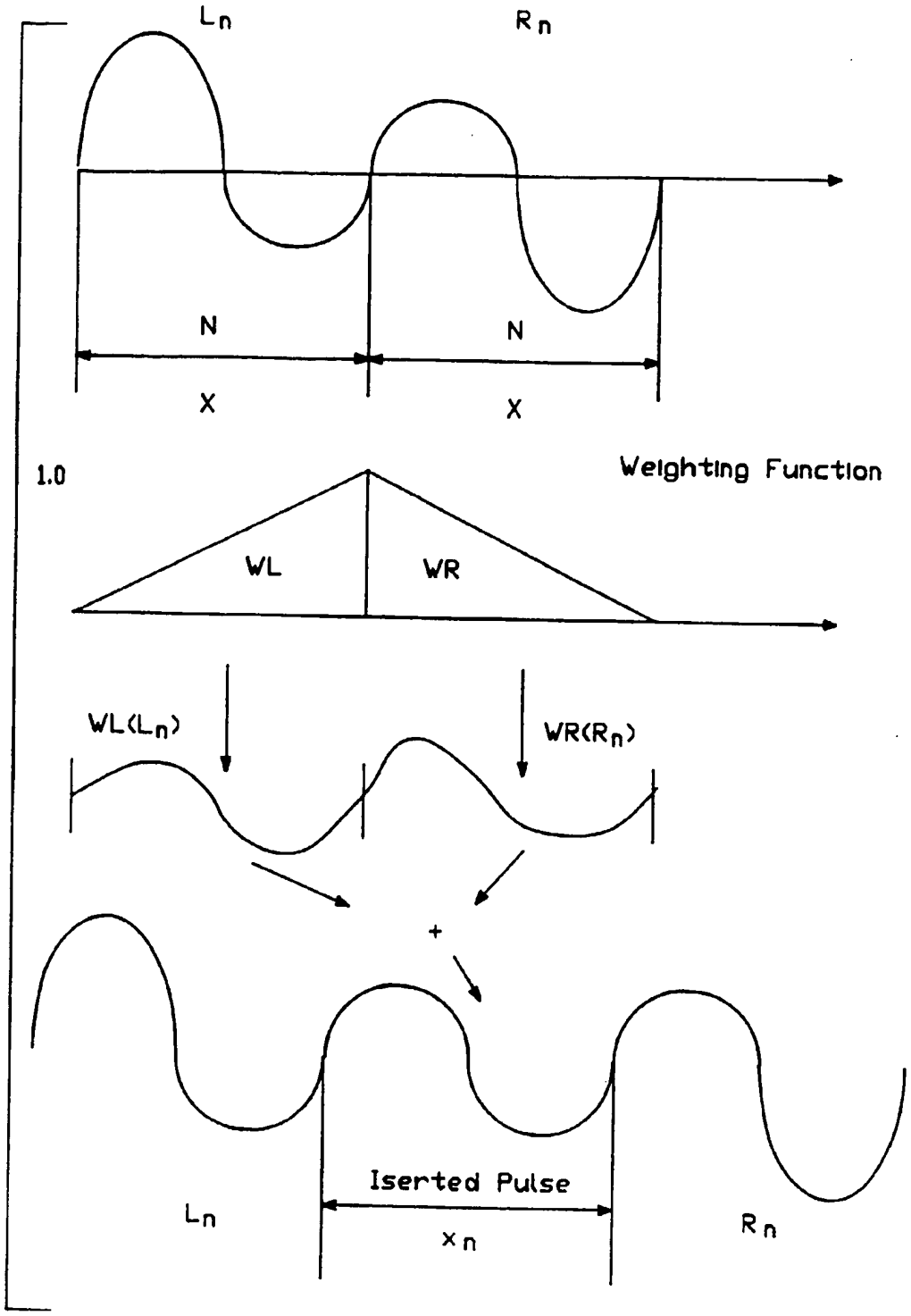

Ln
Rn
N
N
X
X
1.0
Weighting Function
WL
WR
WL(Ln)
WR(Rn)
+
Iserted Pulse
Ln
xn
Rn

FIG.—16

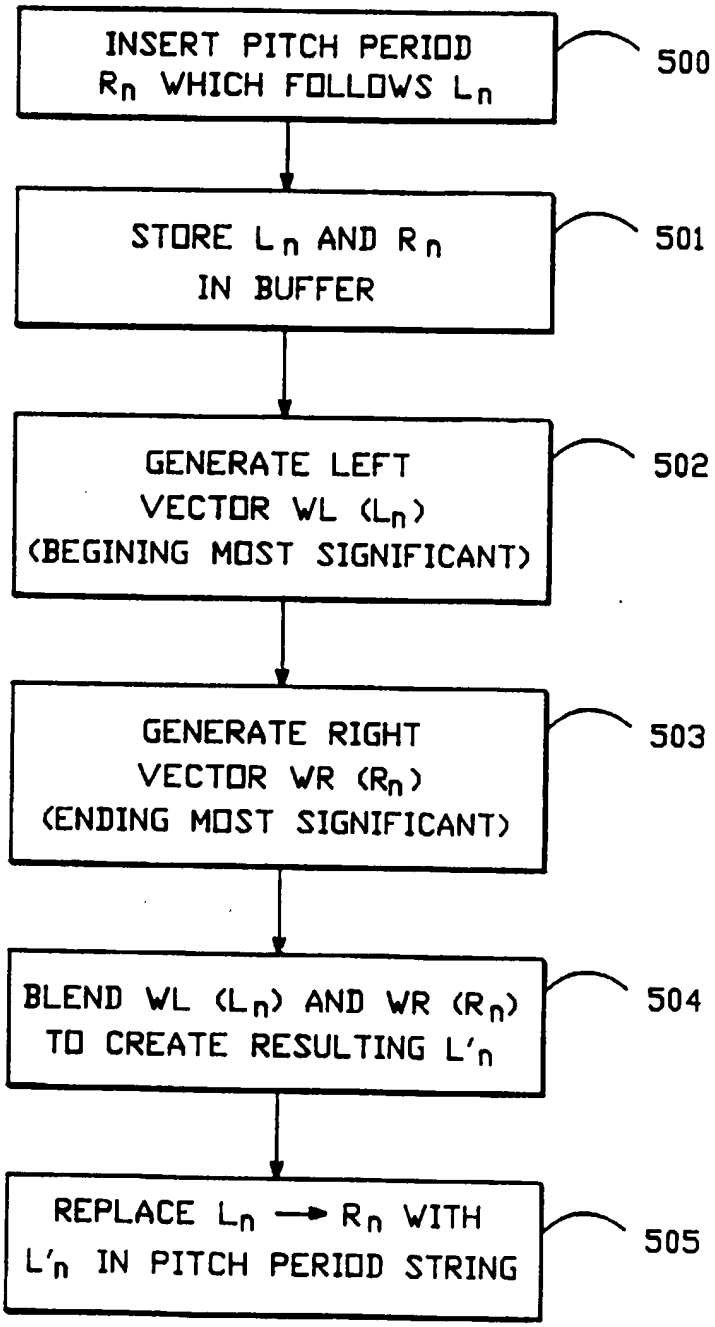

INSERT PITCH PERIOD R_n WHICH FOLLOWS L_n
500
STORE L_n AND R_n IN BUFFER
501
GENERATE LEFT VECTOR WL (L_n) (BEGINING MOST SIGNIFICANT)
502
GENERATE RIGHT VECTOR WR (R_n) (ENDING MOST SIGNIFICANT)
503
BLEND WL (L_n) AND WR (R_n) TO CREATE RESULTING L'_n
504
REPLACE $L_n \rightarrow R_n$ WITH L'_n IN PITCH PERIOD STRING
505

FIG.—17

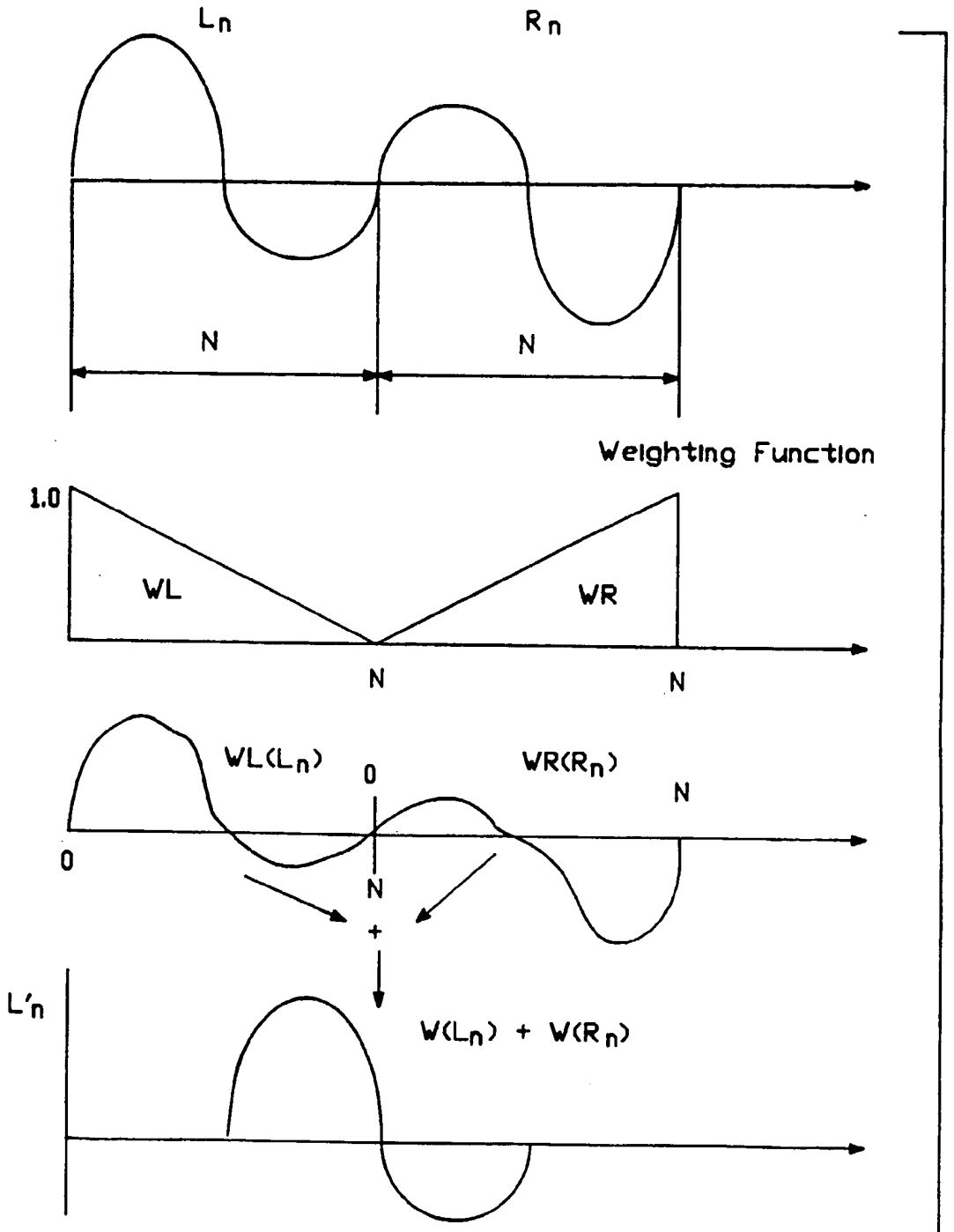

Ln
Rn
N
N
Weighting Function
1.0
WL
WR
N
N
WL(Ln)
0
WR(Rn)
N
0
N
+
L'n
W(Ln) + W(Rn)

FIG.—18